SRI International

R&D Final Report • March 1997

ADVANCED CONTROL SYSTEM DEVELOPMENT AND ANALYSIS

Prepared by:

PAUL J. TITTERTON, *Senior Research Physicist*
Principal Investigator
(415) 859-5619
WILLIAM C. NOWLIN, *Senior Research Engineer*

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

Prepared for:

Office of Naval Research
Code 1222
800 N. Quincy St.
Arlington, VA 22217

Attn: Dr. Geoffrey L. Main
Program Manager, Structures

19970331 094

Contract N00014-94-C-0128
CDRL A0003
ARPA Order A589.14-20
Contract Duration: 31 March 1994 - 31 March 1997

SRI Project 5796

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U. S. Government.

DTIC QUALITY INSPECTED 4

333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326-6200 • FAX: (415) 326-5512 • Telex: 334486

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Summary

The purpose of this project was to develop an advanced control-law synthesis method and apply it to naval noise and vibration problems. Our research was focused along four main paths. Following the Scientific Officer's guidance, we worked to form these paths into a cohesive plan. Our task list comprises

1. Practical robust control design for structures with observed variability
2. Description of variability for robust control law design
3. Cradle characterization experiment design and analysis
4. EM signature work.

The results of these tasks are described below.

The primary result of this contract is SRI's specification-based control-law synthesis methodology and its practical realization in the software package titled `MINCODE` (for `MI`xed `N`orm `CO`ntroller `DE`sign). `MINCODE` synthesizes control laws that directly represent the constraints that occur in real-world fieldable systems. Additionally, `MINCODE` guarantees that this control law yields the best possible performance. No other controller, designed by any other means, will exceed this performance without violating the real-world constraints. This best-performance property enables us to compute optimal tradeoff surfaces for (say) vibration reduction, actuator authority, and stability robustness. These optimal tradeoff surfaces are of obvious utility to the control designer and are unavailable by any other means.

Other results of this contract include: (1) preliminary development of a method for describing plant variability from measured data, (2) preliminary development of a method for incorporating that description of variability into `MINCODE` to improve performance while maintaining robustness, (3) analysis of critical issues in the Cradle characterization experiments, (4) analysis of existing EM signature work, and (5) modeling and simulation of an improved EM signature system.

SRI's preliminary analysis in describing variability and incorporating that description into `MINCODE` indicates that great improvements in vibration reduction can be achieved with little or no practical degradation in stability robustness. Further research is required to refine these techniques and bring them into practical use.

Additionally, SRI's preliminary analysis of variability indicates that current iterative techniques for synthesizing control laws that exhibit *robust performance* can be improved by simple application of `MINCODE`. Direct (as opposed to iterative) synthesis techniques, using `MINCODE` for robust performance, are possible in a limited number of situations. Further research is required to refine these techniques and bring them into practical use.

The EM signature work is continuing under separate sponsorship at a lower funding level.

Supporting work—in the form of the following technical reports, software-usage notes, memos, and published papers—is appended to this report. The description of results provides context for these documents.

- "Two technical explanations: Q-parameter representations and MINCODE norms" by Nowlin, Titterton, and Olkin, 21 Sep. 1996.
- "MINCODE comparison to other control synthesis approaches" by Olkin and Titterton, 18 Aug. 1995.
- "Description of input/output variables for MINCODE" by Olkin and Titterton, 20 Dec. 1995.
- "New MINCODE driver and data builders" by Nixon, 18 Mar. 1996.
- "Performance robustness using MINCODE" by Titterton, Olkin, and Nowlin, Apr. 1996.
- "Practical multi-constraint H^∞ controller synthesis from time-domain data" by Titterton, International Journal of Robust and Non-linear Control. Vol. 6, pp. 413–430, 1996.
- "Semi-definite programming for quadratically constrained quadratic programs" by Olkin and Titterton, SPIE Vol. 2563, pp. 193–204. SPIE Mini-Symposium on Advanced Signal Processing Algorithms, 10 Jul. 1995.
- "Using semi-definite programming for multi-constrained H^2 controller design in active noise control" by Olkin and Titterton, Journal of VLSI Signal Processing, Vol. 14, pp. 57–66, 1996.
- "A practical method for constrained optimization controller design: H^2 or H^∞ optimization with multiple H^2 and/or H^∞ constraints" by Titterton and Olkin, Proceedings of the Twenty-ninth IEEE Asilomar Conference on Signals, Systems, and Computers, Asilomar, CA, pp. 1265–9, 31 Oct. 1995.
- "Selection of neutralization for control-system stability robustness" by Heydt, Dec. 1995.

- "Coupling variability factorization for control-system stability robustness" by Heydt and Nowlin, Apr. 1996.
- "Variability modeling" by Guthart, 20 Mar. 1996.
- Letter to Chris Ruckman at NSWC, 28 Mar. 1995.

Viewgraph briefings and conference presentations are not appended.

Two inexpensive, short-duration tasks were also completed under this program: (i) Airborne Measurements, and (ii) Conversion of LSPS for State Space. In the first, a preliminary set of multi-input multi-output (MIMO) plant data—for use in testing general variability descriptions—was measured in SRI's anechoic chamber. In the second, SRI modified the LSPS software to implement large-scale, state-space descriptions of linear systems.

Practical Robust Control Law Design

This task developed a practical robust control law design methodology and a computer code (`MINCODE`, for `MI`xed `NO`rm `CO`ntroller `DE`sign) that makes such advanced designs tractable. The term "practical" here means that real-world issues such as observed uncertainty in transfer functions, limits on actuator power output, required in-band and out-of-band performance goals, and other constraints are all taken into account. The group developed and implemented a powerful new algorithm based on positive semidefinite programming that enables H^∞, H^2, and ℓ^∞ constrained optimization of control laws for plants with real-world numbers of inputs and output. This implementation also enables constraints using *any combination of these norms.* The result is a control law design approach that directly computes a control law representing the optimization and constraints that occur in real-world fieldable systems. A discussion of the basic parameterization used in MINCODE and the exact meaning of the above norms is included in the appended memo, "Two technical explanations: Q-parameter representations and MINCODE norms," by Nowlin, Titterton, and Olkin, 21 Sep. 1996.

MINCODE is a *significant* advance in the state of the art for MIMO control law design for large (i.e., practical) systems. We are aware of no algorithm in the literature that approaches being able to handle this number of inputs and outputs while still obtaining a rigorously optimal solution in a reasonable amount of time. The code was delivered to NRL in Jan. 1996. The appended memo, "MINCODE comparison to other control synthesis approaches," by Olkin and Titterton, 18 Aug. 1995, discusses MINCODE's place in the realm of controller design techniques.

The code is modular, with H^∞, H^2, and ℓ^∞ modules that are joined at the highest level only. This facilitates the addition of new norm-based constraints, for instance ℓ^1. Two versions of the current code exist: a fast version that enables constraints and an optimization criterion using the H^∞ and H^2 norms, and a slower version that allows the H^∞, H^2, and ℓ^∞ norms. The faster code can solve controller-design problems with a few thousand free variables. Practical considerations and the high-level driver are given in the appended memos, "Description of input/output variables for MINCODE," by Olkin and Titterton, 20 Dec. 1995, and "New MINCODE driver and data builders," by Tom Nixon, 18 Mar. 1996. The MINCODE driver and data builders automate many of the most useful tradeoff studies.

We have demonstrated the ability to trade off controller stability robustness for performance, while maintaining the optimal performance level in every case. Figure 1 is reproduced from a brief, prepared by Paul J. Titterton and William C. Nowlin, entitled "Optimal performance/robustness tradeoff using CTB data with preliminary control approach," dated 18 Apr. 1995. It shows the degree to which performance—as measured in the case of perfect neutralization—degrades as the control law is guaranteed stable for different sized perturbations.

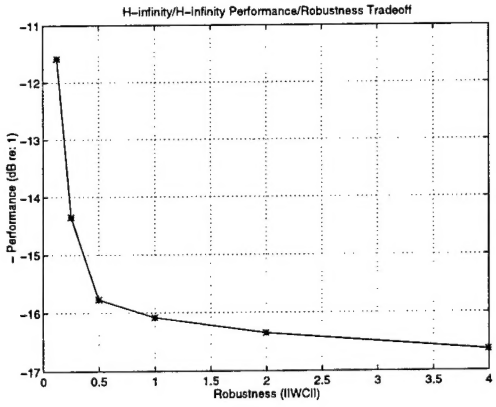

Figure 1: Optimal tradeoff between stability robustness and performance.

The tradeoff is optimal in the following way: At each point on the curve, the coupling is assumed to be perfectly known and is neutralized. An optimal H_∞ design is completed, and its performance using the perfect neutralization of the nominal plant is recorded in dB. The designs of the plants are different, however, as each controller is *designed* to be stably robust to errors in the plant model corresponding to the ∞-norm of the loop gain, $\|WC\|_\infty$. This is the proper measure of stability for an unstructured premultiplicative error in the coupling model. That is, if the product $\|WC\|_\infty\|\Delta C\|_\infty$ is less than 1, then stability is guaranteed. Of course, a similar plot could be produced using the H_2 norm to measure performance.

This plot is the first such ever produced for realistic plants and variability. It shows the best performance one could achieve, under our assumptions of finite impulse response (FIR) filters and H_∞ performance metric, to meet a given requirement of stability robustness. In the case of perfect neutralization, that performance is achieved.

Standard H_∞ methods can produce stable controllers for given levels of variability, but the process is not directly constructive. Instead, the designer must use art and experience to iteratively improve the control design for each point on this curve until it meets the robustness specification. At that point, no guarantees of optimality of the design are obtained, and one is left to wonder whether continued iteration would result in a superior design. As a result, a curve produced by standard H_∞ methods would lie everywhere above the line shown. The performance of the controllers that describe the optimal tradeoff curve is the best it can be while respecting the constraints. SRI's advance in the state of the art of control law design for practical problems takes the guesswork out of synthesis.

It is important to note that Figure 1 shows the best possible performance; performance under situations where neutralization is imperfect will be less (i.e., above the line). The issue of optimizing the performance when the plant model is imperfect is termed "performance robustness" (as opposed to "stability robustness") and is an issue of ongoing study.

Figure 2 shows the degree to which performance—as measured in the case of perfect neutralization—degrades as the control law is guaranteed stable for different sized perturbations. (Fig. 2 is reproduced from a brief, prepared by Nowlin, Titterton, Olkin, Heydt, and Guthart, entitled "SRI ONR Program Update," dated 31 May 1995.) The lower line has been reported earlier, and is the theoretical performance limit, for perfect neutralization, that goes along with guaranteeing stable performance for the indicated maximum percent variability. It is appropriate to ask what the real performance degradation is when true variability is observed while the controller is in operation and the neutralization is no longer perfect.

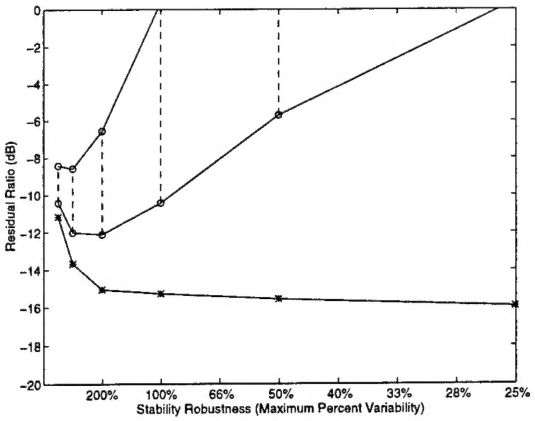

Figure 2: Optimal tradeoff between stability robustness and performance, and resulting observed performance robustness.

The upper two lines show how variability affects performance for this sample problem, in which there are 24 measured plants (one measurement per hour). One plant is taken to be the "design" plant that will be used for neutralization, and the remaining 23 are "test" plants that are used to impose variability into the simulation. That is, we design the controller assuming the "design" plant is the plant in operation, and then deliberately substitute each of the "test" plants in turn in its place, and observe the effect on stability and performance.

The 23 "test" plants have variability on the order of 100%, represented in the units of the abscissa of the plot. Thus, when we use these plants to test control laws that were designed to remain stable only for variability of less than 100%, instability may result. Further, and this is key, the design guarantees only stability, and not any particular level of performance, for a given level of variation. Thus, while use of a particular "test" plant may result in a stable closed-loop system, performance may be a positive number, indicating enhancement rather than cancellation.

The two upper curves represent the performance bounds for all "test" plants that resulted in *stable* closed-loop transfer functions. For the stability robustness designs associated with the left three dotted vertical lines on the plot, all systems possible were stable, and performance fell somewhere along the dotted vertical lines. Toward the right of the plot, using the same plants but different control laws, some plants resulted in unstable systems, and some in systems that enhanced the residual.

As shown on the left of the plot, the control laws designed to be most stable also had the best performance robustness. This correlation is not guaranteed, but seems reasonable. Control laws designed to have less stability robustness are also seen in this example to have less performance robustness. However, it is important to remember that this result, while novel and unique to the program, is anecdotal.

The upper curves highlight the criticality of the performance robustness problem in modern control design. What is not known at this point is how close the performance obtained is to the optimum performance. Obtaining that answer is equivalent to solving the *performance robustness problem.*

In studying performance robustness, we have worked to understand the state of the art, and if possible, apply or extend it using direct controller synthesis and constrained optimization. Additionally, we determined that we can simply adapt "μ-analysis" and "μ-synthesis" to our control design architecture. These tools are widely accepted as the state of the art for *analysis* of performance robustness. Our work toward *direct synthesis* of specified performance robustness using constrained optimization is summarized in the appended technical report "Performance robustness using MINCODE" by Titterton, Olkin, and Nowlin, Apr. 1996.

Much of the work in developing what we call `MINCODE` is devoted to converting the mixed-norm constrained optimization problem to a *semi-definite program,* and exploiting the resulting structure in this program as imposed by the FIR representation and types of constraints applied. This work is extremely technical and its description is contained in the following papers. Conference papers that do not appear in any proceedings are not listed here or appended to this report.

- Titterton, "Practical multi-constraint H^∞ controller synthesis from time-domain data." International Journal of Robust and Non-linear Control. Vol. 6, pp. 413–430, 1996.
- Olkin and Titterton, "Semi-definite programming for quadratically constrained quadratic programs." SPIE Vol. 2563, pp. 193–204. SPIE Mini-Symposium on Advanced Signal Processing Algorithms, 10 Jul. 1995.
- Olkin and Titterton, "Using semi-definite programming for multi-constrained H^2 controller design in active noise control." Journal of VLSI Signal Processing, Vol. 14, pp. 57–66, 1996.
- Titterton and Olkin, "A practical method for constrained optimization controlled design: H^2 or H^∞ optimization with multiple H^2 and/or H^∞ constraints." Proceedings of the Twenty-ninth IEEE Asilomar Conference on Signals, Systems, and Computers, Asilomar, CA, pp. 1265–9, 31 Oct. 1995.

Description of Variability for Robust Control Law Design

All robust control designs necessarily rely on identification of a plant model—whether from measured data or a plant model—and a characterization of system variations. In this task, we worked to answer the two critical questions: (1) what effect does careful choice of a neutralization filter have on system performance and stability for an ensemble of plants, and (2) how is the information from a measured ensemble best used to design constraints to ensure stability over that ensemble?

A comprehensive analysis of the neutralization-selection problem is in the appended technical note, "Selection of neutralization for control-system stability robustness" by Heydt, Dec. 1995. This note documents recent results on the subject of control system performance in the presence of variability. Coupling variability must be accounted for in controller design if stability is to be maintained, and this exacts a price on system performance. The coupling neutralization was chosen as part of the design process, and an important issue was to determine what effect, if any, the choice of neutralization had on performance. This issue was investigated by designing example controllers based on the CT-B data set (24 hours' worth of coupling data) and taking into account the observed measurement variability. The effect on performance was illustrated for three choices of coupling neutralization, and the results show that proper selection of neutralization can mean the difference between marginal performance (3 dB) and good performance (> 8 dB). The work reported in this note expanded on preliminary work on the tradeoff of control-system performance and stability robustness by Titterton and Nowlin presented in the briefing, "Optimal performance/robustness tradeoff using CT-B data with preliminary control approach," on 18 Apr. 1995.

The neutralization selection problem discussed in Heydt's technical note is properly formulated as a convex program. As such, it is readily solved using SRI's mixed norm constrained optimization program, the `MINCODE` engine, showing that `MINCODE`'s applicability is not limited to the final stage of control law design. We designed a 2×2 neutralized feedforward controller with the CT-B data and made predictions of performance vs. stability-robustness, using an additive model of coupling variability. With the additive model, control performance improves as the maximum infinity-norm radius decreases. We addressed the question of how important the choice of the nominal coupling (N) is to control-system performance. Our results show that performance may be improved significantly (several dB for this data set) if the nominal coupling is determined via optimization, rather than by simply selecting a "typical" coupling measurement.

The appended technical note, "Coupling variability factorization for control-system stability robustness" by Heydt and Nowlin, Apr. 1996, and the appended memo, "Variability modeling" by Guthart, 20 Mar. 1996, both describe a single-input, single-output (SISO) specialization of a MIMO convex problem that will frequency-weight the stability-robustness constraint presented to `MINCODE`. The weighting contours the constraint as a function of frequency, looser where the identification is good, and tighter where the identification is poor and greater robustness is required. This contoured constraint, which describes the structure in the variability, produces much better performance than the uncontoured constraint, which assumes an unstructured variability.

As the program ended, we were investigating methods to speed the calculation of optimal MIMO plant models and their structured variability. The key to this investigation is the required causality of the variability's structure. If the structure must be described by causal transfer functions, then the required computations will be of the same complexity as MINCODE. If the structure can be described by the transfer function's correlation matrices, then the required computation will be an order of magnitude less than MINCODE.

Because of a lack of funding, the automatic description of variability from measured data is not yet possible for MIMO plants.

Cradle Characterization Experiment Design and Analysis

In this task, SRI advised NSWC on the Cradle experiment design, as far as this experiment was supposed to shed light on active control. Additionally, SRI commented on proposed controller notions. We never received the data, and therefore were never able to perform any analysis. An example of our advice is appended in the 28 Mar. 1995 letter to Chris Ruckman at NSWC.

Our charter in this work was twofold: (1) continue to validate our control law design methodology on realistic structures and (2) investigate the critical issues for active isolation of the proposed Cradle structure. To validate the active isolation approach, we tried to specify sensors, actuators, and their mount points, along with measurement specifications for the ISMS test. Effective isolation depends critically on the determination of the physical mechanisms by which acoustic energy is transferred to the surrounding medium. Our goal is to predict those mechanisms and verify them during the ISMS test.

Since the control system is being designed from the ground up, design issues such as transducer design and placement would have needed to be evaluated in light of propagation paths through the fluid-loaded structure. As a first step, we would have evaluated the effect of actuators placed off the passive mounts on a combined isolation/PSI control system having a fluid-loaded object with an internal substructure. Preliminary numerical experiments demonstrated that noncollocated actuators will launch structural waves onto the object above a frequency that depends upon structural waves in the object. Further, we demonstrated that PSI and isolation control will share control algorithms and design issues whether or not they are implemented simultaneously. This would have allowed us to leverage prior work on PSI control for application to the isolation objective. For example, issues of coupling and interconnection, explored extensively in the PSI problem, would have arisen again in the isolation objective. We planned to use the framework developed for analyzing these issues to investigate the following:

- Interconnection issues. Coupling in the substructure and through the object would have determined the connections between sensors and actuators. For a highly damped structure, purely local interconnections (SISO) may have been sufficient. The physics of the coupling would need to be investigated to determine the required control network.
- Transducer type and placement. The mechanisms governing transmission of vibrations through the mounts would need to be explored in order to determine the best transducers for the control system.

After the test, the answers to these questions would have guided SRI in the design and accurate simulation of a specific control law that could be realized in hardware. Guarantees of stability and performance in the presence of observed transfer function uncertainty would have been a primary concern.

EM Signature Work

The EM signature work was divided into two tasks. The first analyzed an open-loop controller design approach, and the second analyzed a closed-loop control approach. The open-loop approach was disposed of rapidly.

SRI completed analysis of closed-loop stability of a system based on moment correlation or matrix-of-states approaches using simple analytical dipole models and lumped circuit models for the actuators. This analysis showed how instabilities can occur in a multichannel system due to miscalibration of actuator locations.

SRI developed and demonstrated a closed-loop control design methodology, based on H^2 optimal control, on simulated data using triaxial dipole models. Performance, stability, and robust performance in the presence of miscalibrations were demonstrated via numerous examples. The final report for the EM Signature Work, "Closed-loop signature control task final report" by Chou is available from Paul Izat (410 293 9524) at NSWC.

ATTACHMENTS

Memo AC&SPG

TO: Dr. Geoffrey L. Main
Office of Naval Research
Program Manager, Structures

FROM: William C. Nowlin, Paul J. Titterton, Julia A. Olkin

DATE: September 21, 1995

SUBJECT: Two technical Explanations: Q-parameter representations and MINCODE norms

Introduction

This brief memo goes into the detail you asked for when you visited SRI on 18 August, 1995. In it, we answer two separate questions.

1. Does the Q parameterization work for representations other than FIR models, for example IIR or state space representations? And what is the applicability of MINCODE in these cases?

2. What is the exact meaning of the different norms implemented in MINCODE? What is the difference between H_∞ and ℓ_∞, for instance?

Summary

It is possible, in each of the three cases (FIR, IIR, and state space) to create a Q parameterization where the closed-loop transfer matrix is linear-plus-offset ("affine") in Q. However, for the approach to be useful, Q must parameterize the entire space of stabilizing controllers, and its representation (FIR, IIR, or state space) should be complete and yield compact representations for real-world plants.

Additionally, the design problem that remains is challenging, in that we must solve a mixed norm constrained optimization problem in the free variables of the particular representation. This problem is nonlinear for the poles of an IIR or state-space representation. The FIR representation yields a linear dependency on the free parameters that enables the use of fast interior point methods to satisfy mixed-norm constraints. Nevertheless, if the pole placement problem can be addressed somehow, then a mixed FIR-IIR representation could be very efficient and beneficial to the program. MINCODE is optimized to solve the FIR case, and would need to be rewritten to take advantage of other structure. Further discussion of these points is given in Section 1.

333 Ravenswood Ave. • Menlo Park, CA 94025 • (415) 326-6200 • TWX: 910-373-2046 • Telex: 334486 • Facsimile: (415) 326-5512

Here are verbal descriptions of the four MINCODE norms. See Section 2 for the underlying mathematical explanation.

- The H_∞ norm measures the worst case RMS gain of a stable MIMO transfer matrix. It is the worst-case RMS value of any multichannel output signal as the input varies over all signals with unit RMS value. It is used for stability robustness constraints and also to minimize the worst case disturbance sensitivity of the system.
- The H_2 transfer matrix norm is the RMS value of the multichannel output signal when the inputs are driven by independent white noise. It is used primarily to minimize the RMS response of the system for white or colored noise disturbances.
- The ℓ_∞ norm measures the largest instantaneous value over all output channels of a transfer matrix for a finite set of known input signals. It is used to ensure that instantaneous actuator voltage limits are not exceeded for a known disturbance to the system.
- The ℓ_1 transfer matrix norm is the worst-case maximum amplitude of any output signal given any input signal of unit maximum amplitude. It is used to ensure that instantaneous actuator voltage limits are not exceeded for any disturbance of known bound.

1 FIR, IIR, and State-Space Q Parameters

It is reasonable to ask about the usefulness of the Q parameterization when the plant and/or controller transfer functions are described by models other than FIRs, for example state space or IIR representations. In past discussions we have mentioned that the answer is a little complex. In this section, we offer an explanation.

The Q parameter is a MIMO transfer function parameterization of the controller. There are four requirements for a successful Q parameterization.

1. Break the controller, K, into two parts: an initial stabilizing controller, which does not give the final desired closed-loop response, and Q, upon which the closed-loop transfer function (CLTF) dependence is *linear plus offset* (or *affine*).
2. Show that the representation is complete (i.e., that as Q ranges over all stable transfer functions, the resulting CLTF ranges over all stable transfer functions).
3. Discretize Q so that it may be numerically computed, ensuring that the coefficients converge rapidly. The best discretization depends on the type of plant being controlled.
4. Show that the optimization of the resulting free parameters in Q can be solved in an efficient manner.

There are two well-known forms of the Q parameterization: neutralized feedforward (Figure 1(a)) and linear quadratic regulator-linear quadratic estimator (LQR-LQE) (Figure 1(b)). Both generate stabilizing controllers. Below is a brief comparison of the two forms of the stabilizing controller and the requirements of each.

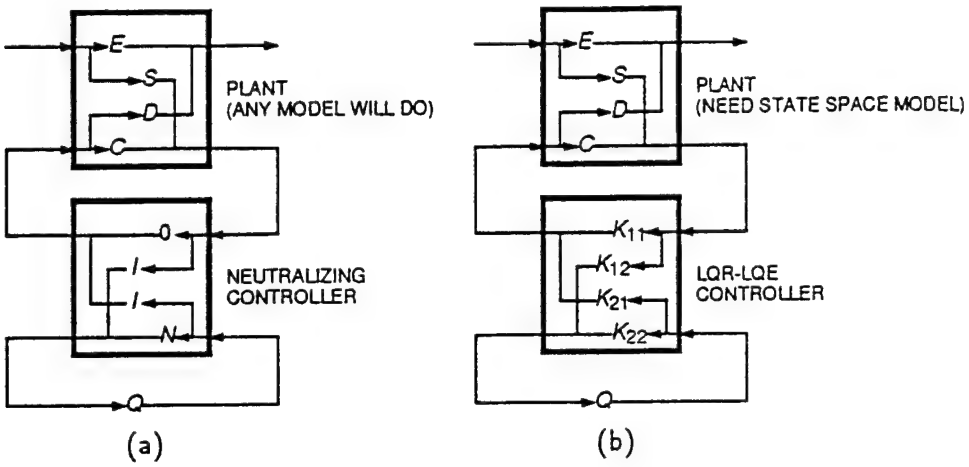

Figure 1: (a): "Neutralized Feedforward" Q parameterization: the symbol *I* represents the identity transfer matrix, and 0 represents the zero transfer matrix; (b): "LQR-LQE" Q parameterization.

	Neutralized Feedforward	LQR-LQE
Plant model	Any model	Need state space model
Computation	N given by direct measurement of C	Calculated by state space methods
Implementation	Any representation	Any representation
Caveats	Plant must be stable	Plant may be unstable

For computer-aided design using interior point methods (or any other fast minimization approach), we require that the transfer matrix Q satisfy

$$Q = \sum_m Q_m \alpha_m$$

where α_m are the free parameters and Q_m are some set of stable transfer functions. The sum must be complete, and must converge quickly enough to be of practical utility. If Q is described in this way, then *any norm on the CLTF is convex in the* α_m. This together with fast convergence yields a numerically tractable optimization approach.

We turn now to the three candidate representations for Q.

1.1 FIR Representation

Here

$$Q = \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{n=0}^{N-1} Q_{m,k}(n) \alpha_{m,k,n}$$

where $Q_{m,k}(n)$ is just a basis function representing a unit impulse at the (m,k) crosspoint delayed by n samples (i.e., a basis function corresponding to a single tap). Similarly, the free parameter, $\alpha_{m,k,n}$, is just the numerical value of that tap.

This decomposition is complete as $N \to \infty$.[1] In our control problems, we generally require $N \lesssim 1000$, which, given the inherent structure of the FIR representations of the system and Q parameter, leads to a tractable design problem.

1.2 IIR Representation

In this case, we have the form

$$Q = \sum_{m=1}^{M} Q_m \alpha_m \qquad \text{where} \qquad Q_m = \frac{z}{z - \lambda_m}$$

This form is complete for some choices of λ_m as $M \to \infty$, but the λ_m must be determined *a priori* by some method, as they appear nonlinearly in the CLTF.

If these parameters are determined by some means, then the α_m may be efficiently optimized using interior point methods, including the imposition of mixed-norm constraints. Currently we are aware of no method to choose optimal $\{\lambda_m\}$. Nevertheless, we believe this representation is worth further study, because often the long-time response of the Q-parameter appears to be a sum of a few decaying exponentials. Much computation could be saved if these were modeled as such and not as FIRs, although this would require a hybrid FIR/IIR representation for Q.

1.3 State-Space Representation

The state space representation is a notationally compact representation for linear combinations of FIR and IIR filters. It has the same properties and pitfalls. In particular, using the standard (A_Q, B_Q, C_Q, D_Q) state-space notation for Q, if either A_Q and B_Q are held constant or A_Q and C_Q are held constant, then the normed CLTF,

$$\left\| E + D\left[C_Q(Iz - A_Q)^{-1}B_Q + D_Q\right]S \right\|$$

is convex in the remaining free parameters.

1.4 Applicability of MINCODE

`MINCODE` takes advantage of special structure presented by the FIR representation of the plant model and Q parameter. Thus, it is not optimized for other representations, for which the structure is different. However, such structure could be exploited by writing additional code.

1.5 Interchangeability of Representations

Finally, it is important to note that, regardless of how a control law is computed, it is possible to translate it from one representation to another. For example, it is possible to compute an advanced controller using MINCODE and an FIR representation, and implement that controller using state-space hardware. Such a conversion could entail writing

[1] See Boyd & Barratt, *Linear Controller Design: Limits of Performance*, Prentice-Hall, 1991, p. 160.

the FIR design in a special state-space format and then performing a model-order reduction to reduce the state-space representation to an approximately equivalent but more compact one which could be implemented in available hardware. SRI has accomplished this for modest problems to test the reconfiguration of its Advanced Signal Processor (ASP) as a state-space controller. Alternately, one could perform system identification on the frequency response of the FIR. Other approaches are also possible.

2 Definitions and Examples of Different Norms

Here is a brief description of the different norms that are implemented in `MINCODE`. We have added historical notes where appropriate. A complete discussion of the theory of signal and operator norms is beyond our needs. More details are given in Boyd & Barratt, pp. 110–126, and in Desoer & Vidyasagar,[2] pp. 10–26.

2.1 Underlying Machinery of MINCODE Norms

There are layers of norms in this type of work. To make what we use as clear as possible, we will devote a brief portion of this note to the underlying machinery that we must use. In particular, we will define

- Vector spaces
- Norms on vector spaces
- Vector-valued signals
- Norms on vector-valued signals
- Operators, and the space of operators
- Operator norms and induced operator norms

Def 1 *A* real vector space *is a set, V, whose elements are called vectors and in which vector addition and scalar multiplication are defined and closed. There exists a unique vector called zero, vector addition is associative, and scalar multiplication is distributive.*[3]

Our use of vector spaces will be confined to the familiar $\mathbf{R}^M$ for positive integer M: the set of M-tuples of real numbers. The m^{th} entry of the M-tuple x is denoted x_m.

Def 2 *A* norm *on a vector space V is a function defined on that space that associates with each vector x a real number $\|x\|$ satisfying the following properties:*

(i) The norm is always non-negative, and is zero only for the zero vector;

(ii) $\|\alpha x\| = |\alpha|\|x\|$ for all $\alpha \in \mathbf{R}$ and $x \in V$;

[2] Desoer & Vidyasagar, *Feedback Systems: Input-Output Properties*, Academic Press, 1975

[3] For a precise definition, see, e.g., Rudin, *Real and Complex Analysis* third edition, McGraw-Hill, 1986, p. 33.

(iii) $\|x+y\| \leq \|x\| + \|y\|$ *for all* $x, y \in V$ *(triangle inequality).*

Obviously there is an infinite variety of norms on $\mathbf{R}^M$. Important ones for this work are the p-norms, defined as

$$\|x\|_p \triangleq \left[\sum_{m=1}^{M} |x_m|^p\right]^{1/p} \qquad \text{(vector p-norm)}$$

for integer $1 \leq p < \infty$. In some ways the most "natural" choice is the Euclidean norm, $p = 2$, because that norm gives the *distance* from the origin in a cartesian coordinate system with M coordinates. However, for our applications, other norms are more well-suited to constraining closed-loop responses of systems.

A particularly important limiting case of the p-norm is the ∞-norm, defined as

$$\|x\|_\infty \triangleq \max_{1 \leq m \leq M} |x_m| \qquad \text{(vector } \infty\text{-norm)}$$

Def 3 *A* discrete-time vector-valued signal *of dimension M is a function, u, mapping the non-negative integers to* $\mathbf{R}^M$*:*

$$u(\cdot): \{0\} \cup \mathbf{Z}^+ \rightarrow \mathbf{R}^M$$

Included in this general class of signals are signals of both finite and infinite duration. Signals may take values, $u(n)$, that are unbounded as $n \rightarrow \infty$, but may not take infinity as a value.

Def 4 *For $1 \leq p < \infty$, the functional $f_p(\cdot)$ acting on discrete-time vector-valued signals in* $\mathbf{R}^M$ *is defined as follows:*

$$f_p(u) \triangleq \left[\sum_{n=0}^{\infty} \|u(n)\|_p^p\right]^{1/p}$$

For $p = \infty$:

$$f_\infty(u) \triangleq \max_{0 \leq n < \infty} \|u(n)\|_\infty$$

Def 5 *For $1 \leq p \leq \infty$, the space of discrete-time $\mathbf{R}^M$-valued signals for which $f_p(u)$ is bounded is denoted ℓ_p^M. In this case we define the signal norm as*

$$\|u\|_{\ell_p} \triangleq f_p(u) \qquad \text{(signal p-norm)}$$

The signal p-norm defined on ℓ_p^M is thus the natural norm, and in the literature is called the ℓ_p signal norm. Note that the signal p-norm always uses the vector p-norm as its underlying vector norm. This holds for the remainder of this memorandum and for MINCODE.

Def 6 *An* operator, $\mathcal{Q}$, *mapping ℓ_p^K to ℓ_p^M is a linear map*[4] *of each signal $u \in \ell_p^K$ to a signal $\mathcal{Q}(u) \in \ell_p^M$.*

[4] A space of linear maps defines addition of two maps and multiplication of a map by a scalar. Also two linear maps may be composed in the obvious way if the range of one is the domain of the other. Note that for maps from ℓ_p^M to ℓ_p^M, the composition does not in general commute. See Desoer, p. 20.

A convenient method of representing linear, causal, shift-invariant,[5] stable[6] operators from ℓ_p^K to ℓ_p^M is via convolution with an impulse response signal, with convolution denoted by "$*$". Thus we write the output of the operator as

$$\mathcal{Q}(u) = Q * u$$

where Q is the impulse response matrix. For the case of interest, we model the impulse response as an FIR and represent it as a collection of tap weights, each written $Q_{m,k}(n)$, where $1 \leq m \leq M$ and M is the number of output channels, $1 \leq k \leq K$ and K is the number of input channels, and $0 \leq n < N$ and N is the number of time samples. Q may be thus thought of as a matrix-valued signal. Notationally, we write $\mathcal{Q}$ for the operator, Q for its matrix impulse response, and $\hat{Q}$ for its matrix transfer function (which is related to Q by the Fourier transform).

In general a vector space may have a variety of norms available. Operators form a vector space; some norms on this vector space relate to the properties of the vectors as operators, and others do not. It is often beneficial to discuss a norm's behavior by identifying it as an *induced operator norm*, so that the relationship between the operator's action and its norm is clear.

Def 7 *Let $\mathcal{Q}$ be an operator mapping the normed space A to the normed space B. Then the* induced operator norm *of $\mathcal{Q}$ is*

$$\|\mathcal{Q}\|_{\text{induced}} \triangleq \sup_{\|u\|_A=1} \|\mathcal{Q}(u)\|_B \qquad \text{(induced operator norm)}$$

With these general definitions in mind, let us examine the four norms currently implemented in `MINCODE`. In each case we assume the operator $\mathcal{Q}$ is mapping K inputs to M outputs, so that $Q(n) \in \mathbf{R}^{M \times K}$.

2.2 The H_∞ Operator Norm

This norm is defined for the operator space $H_\infty^{M \times K}$, which is the space of linear, causal, shift-invariant, stable transfer functions. Although the following is properly presented as a theorem, for our purposes we may take it as a definition of the H_∞ norm: The H_∞ operator norm is the operator norm induced by the ℓ_2 signal norm. That is,

$$\|\mathcal{Q}\|_{H_\infty} \triangleq \max_{\|u\|_{\ell_2}=1} \|Q * u\|_{\ell_2} \qquad (H_\infty \text{ norm})$$

With this "definition" in hand, it is clear that *the H_∞ norm measures the worst case RMS gain.* It is the worst-case RMS value of any multichannel output signal as the input varies over all signals with unit RMS value.

For completeness, we note that this norm equivalently computes the supremum over $z \in \mathsf{T}$ of the largest singular value of the matrix frequency response (T is the unit circle

[5] Consider the delay operator, δ, which delays a signal by 1 sample. $\mathcal{Q}$ is defined to be shift invariant if $\mathcal{Q}(\delta(u)) = \delta(\mathcal{Q}(u))$ for all $u \in \ell_p^K$.

[6] I.e., bounded: there exists a constant, $k_{\mathcal{Q}}$, such that for all $u \in \ell_p^K$, $\|\mathcal{Q}(u)\|_{\ell_p} \leq k_{\mathcal{Q}} \|u\|_{\ell_p}$.

in the complex plane). For computation, T is sampled at a finite number of points, $\{z_n = \exp[2\pi nj/N]\}_{n=0}^{N-1}$. Thus, in MINCODE we approximate the computation by

$$\|\mathcal{Q}\|_{H_\infty} = \max_{0 \le n < N} \overline{\sigma}\left(\hat{Q}(e^{\frac{2\pi nj}{N}})\right)$$

2.3 The H_2 Operator Norm

The H_2 norm is defined for the space of operators that act by convolution, denoted $H_2^{M\times K}$. The discrete time definition of the norm is

$$\|\mathcal{Q}\|_{H_2} \triangleq \frac{1}{2\pi}\left[\int_{-\pi}^{\pi} \operatorname{tr} \hat{Q}^H(e^{j\omega})\,\hat{Q}(e^{j\omega})\,d\omega\right]^{1/2} \qquad (H_2 \text{ norm})$$

where $Q^H(z)$ is the conjugate transpose of $Q(z)$. Using Parseval's theorem and assuming the impulse response is finite yields the representation used in MINCODE:

$$\|\mathcal{Q}\|_{H_2} = \left[\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=0}^{N-1}|Q_{m,k}(n)|^2\right]^{1/2}$$

In this case the physical interpretation is a statistical (average) one rather than one for the deterministic worst case: *The H_2 norm of $\mathcal{Q}$ is the RMS value of the output when the inputs are driven by independent white noise.*[7]

We note this is the same norm that is employed in SWAPS, for both the optimization and the single constraint that it can apply. Thus, inclusion of this norm means that MINCODE is a superset of SWAPS. Additionally, although the interpretation of the H_2 norm is in terms of white inputs, the inputs can be frequency-weighted to highlight certain frequency bands, as is often done in a SWAPS preprocessing step.

2.4 The ℓ_∞ Norm

The norm implemented in MINCODE called "ℓ_∞" is actually a signal norm, rather than an operator norm. Thus, if $u(n) \in \mathbf{R}^M$ for each n (with the norm on $\mathbf{R}^M$ being $\|\cdot\|_\infty$), and assuming the signal's impulse response is of finite length N, the definition is

$$\|u\|_{\ell_\infty} \triangleq \max_{0 \le n < N} \|u(n)\|_\infty$$

Thus the ℓ_∞ norm measures the largest instantaneous value of any one channel of a signal. The reason this norm is implemented as a signal norm is that it is used to constrain the convolution of the controller weights, W, and the input signal, $S * e_{\mathrm{ex}}$, where e_{ex} is a *known* excitation signal (i.e., a single vector signal rather than a space of vector signals) and S is the excitation-to-sensor-output transfer matrix.

Such a constraint indirectly affects the weights, directly ensuring that the control commands never exceed certain levels for one input signal. By taking the maximum of many such norms, one for each of a finite set of known inputs, we can ensure that the

[7] Boyd & Barratt, p. 110

control commands never exceed specified levels for any of the given input signals. The norm that is constrained in MINCODE is thus

$$\left\|\{u_\alpha, u_\beta, \ldots, u_\omega\}\right\|_{\ell_\infty} \triangleq \max_{0 \le n < N_n} \left[\max\left\{\|u_\alpha(n)\|_\infty, \|u_\beta(n)\|_\infty, \ldots, \|u_\omega(n)\|_\infty\right\}\right] \quad (\ell_\infty \text{ norm})$$

2.5 The ℓ_1 Norm

The final norm, which will be added to MINCODE shortly, is the operator norm induced by the ℓ_∞ signal norm with underlying vector ∞-norm. It is sometimes called the ℓ_1 operator norm, because in the scalar case it is equal to the ℓ_1 norm of the impulse response. We have

$$\|\mathcal{Q}\|_{\ell_1} \triangleq \max_{\|u\|_{\ell_\infty}=1} \|\mathcal{Q}(u)\|_{\ell_\infty} \qquad (\ell_1 \text{ norm})$$

Thus the ℓ_1 operator norm is the worst-case maximum amplitude (worst over both channels and time samples) of any output signal for all input signals of unit maximum amplitude. This is exactly what is required to constrain the control command voltage for all excitation signals with unknown shape but known peak value.

For completeness, we note that, assuming the impulse response Q is finite, the ℓ_1 operator norm may be computed as[8]

$$\|\mathcal{Q}\|_{\ell_1} = \max_{1 \le m \le M} \sum_{k=1}^{K} \sum_{n=0}^{N-1} |Q_{m,k}(n)|$$

[8] Desoer & Vidyasagar, p. 26

MEMO

To: Dr. Geoff Main

From: Julia Olkin and Paul Titterton, Jr.

Date: August 18, 1995

Subject: *MINCODE comparison to other control synthesis algorithms*

- **Motivation for developing MINCODE:**
  1. We're unable to use any algorithms based on state space methods because no accurate models exists for our plant
  2. We're unable to use any other existing algorithms in the field because they do not take advantage of inherent problem structure, and thus, cannot handle the large, realistic problem sizes we require.
- **MINCODE:** Capabilities Overview
  1. Direct translation of specifications to optimum control law
  2. Uses measured impulse response data directly
  3. Mixed norm (H_2, H_∞) optimization with multiple mixed-norm (H_2, H_∞, l_∞) constraints
  4. Implements optimization/constraints for
     - Plant dynamics constraints
       * Out-of-band enhancement
       * Out-of-sector enhancement
     - Robustness
       * Stability robustness
       * Performance sensitivity
       * Performance robustness
     - Physical constraints
       * Actuator authority limits
       * Sensor input limits
  5. Handles large (realistic) problems efficiently
- **MATLAB:**
  1. Optimization Toolbox

   - General constrained optimization problem; does not take any advantage of inherent structure in problem
   - Would work for small problems. But our required problem sizes are much too large: we can not store the Hessian in full form as optimization routines would require.

2. Control Toolbox
   - Can only do LQG (H_2) controllers.
3. Robust Control Toolbox and Mu Analysis Toolbox
   - Can only do either H_2 or H_∞ controllers with *no* constraints.
   - Like SWAPS without Tikhonov; get regularization by augmenting plant.
   - Advantage: does mu analysis whereas MINCODE doesn't. Useful in bounding performance-robustness, although is sub-optimal.
   - However, doesn't have variety of norms or the norm mixing as in MINCODE. Could not constrain actuator authority or out-of-band while optimizing performance criterion.
4. Linear Matrix Inequality (LMI) Toolbox (brand new)
   - We solve an LMI problem using Semi-Definite Programming (SDP). SDP: linear cost with linear matrix inequality as feasible set. Presumably, could solve small problems with this toolbox.
5. Above four related control Toolboxes all use state-space (SS) controllers, although LMI could be adapted to Q-parametrization for small problems.
6. SS models can be achieved by analytic methods, but no accurate analytic models exists for our plants.
7. SS models achieved by system identification
   - Expect it to take hundreds of states to reproduce dynamics
   - Expect derived model does not reflect real plant well; no clear pole/zero representation of plant.
8. Even if SS fell into place, would *still* not allow all the cost functions we require.

- **Functionality:**
   - We want to go from specifications to controller design. That requires us to be able to simultaneously treat the feasibility problem (requires norm mixing) and optimization problems (generally H_2 or H_∞). Cannot solve this problem directly with the size we require with anything but MINCODE.

- There are several differences between MINCODE's capabilities and anything else

1. MINCODE: Mixed-norm capability (H_2 and H_∞ objective with H_2, H_∞, and L_∞ constraints)

2. MINCODE: Go directly from specifications to control synthesis.
3. Plant Model
   - MINCODE uses impulse response measurement for plant so use measured data directly versus SS representation
   - For acoustic radiation, SS is *not* more compact
   - No modeling step or system ID step is needed
4. Controller Parametrization
   - Neutralized feedforward parametrization is the simplest Q-parametrization (or Youla parametrization). Our parametrization for Q is different: MINCODE uses FIR versus IIR for computational efficiency and completeness (all possible Q parameters upto whatever length can be represented by that FIR).
   - MATLAB routine called "youla:" gives LQR/LQE-type parametrization (versus MINCODE's). LQR finds state feedback gains and LQE (least quadratic estimator) finds Kalman filter gains. Do not need neutralization but need an exact SS model of plant for it to work. In effect, need to develop stabilizing controller which becomes neutralization.
5. Numerical Algorithm
   - SS methods revolve around solving coupled Ricatti equations. The developed controller has two matrix parameters to solve for, arising from Ricatti equations.
   - We translate to SDP and solve using interior-point methods.

- **Differences in application:**

1. MINCODE (convex optimization):
   - Will either synthesize a controller to meet the specifications, or know that the specs are infeasible.
   - No iterations required. Must have convex specifications in order for this to work, but most of our specs can be posed as convex.
   - Other convex optimizers cannot handle real-world problem sizes.
2. Other methods:
   - Take all closed-loop transfer functions (CLTF) of interest, and incorporate into single H_2 or H_∞ minimization problem. Solve that problem, with filters, weightings, etc. Then go back and simulate whether controller meets specifications. If not, then change weighting factors.
   - Iteratively must change weightings until specifications are met, or until you give up. Never sure if you couldn't do better by wiser choice of weightings.

MEMO

To: Users of MINCODE

From: Julia Olkin and Paul Titterton, Jr.

Date: December 20, 1995

Subject: *Description of Input/Output Variables for MINCODE*

1 Introduction

This memo gives a detailed description of the input and output variables used in MINCODE. In order to understand some of the data descriptions, it is important to know the nomenclature used in the problem setup. The control problem which this particular version of MINCODE solves is

$$\begin{array}{ll} \min\limits_{\mathbf{W}} & \|\mathbf{E}_0 + \mathbf{D}_0 \circledast \mathbf{W} \circledast \mathbf{S}_0\|_{p_0} \\ \text{subject to} & \|\mathbf{E}_j + \mathbf{D}_j \circledast \mathbf{W} \circledast \mathbf{S}_j\|_{p_j} \leq \gamma_j,\ j = 1, \ldots, k, \end{array} \tag{1}$$

where the norm $p_j, j = 0, ..., k$ can be either H_∞ or H_2, and $\circledast$ is the convolution operator. MINCODE incorporates semi-definite progamming techniques, in particular, a primal-dual potential reduction method. We translate the control problem into a particular semi-definite program as follows,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|\mathbf{E}_0 + \mathbf{D}_0 \circledast \mathbf{W} \circledast \mathbf{S}_0\|_{p_0} \leq t, \\ & \|\mathbf{E}_j + \mathbf{D}_j \circledast \mathbf{W} \circledast \mathbf{S}_j\|_{p_j} \leq \gamma_j,\ j = 1, \ldots, k. \end{array} \tag{2}$$

The calling sequence for MINCODE is as follows:

```
[t, W, Z_hi, a_z, N_z, A_z, alpha, infor, history] = ...
      Mincode_hih2(E, edex, S, sdex, D, ddex, con_type, ...
              gamma_hi, gamma_h2, nf, nt, ns, nd, nr, ...
              t0, W0, Z_hi0, a_z, N_z, A_z, ...
              Mt, MW, nu, tol_nd, tol_ps, maxit_nd, maxit_ps);
```

2 Input Variables

The input variables can be divided into three classifications: (i) variables which describe the problem to solve and give data information, (ii) the initial primal and dual variables, and

(iii) variables particular to the algorithm and not the problem itself. The input variables in `Mincode_hih2` are ordered along these classifications, and are discussed in more detail below.

2.1 Specifications of Problem and Data

E,S,D: Contains the E, S, and D data for *all* constraints and minimization criteria. Each column is a time-series. Assume that the length of E is the defining length. Thus, the length of E must be greater than the length of the convolution of D with W with S. (i.e. $E_{len} > D_{len} + W_{len} + S_{len} - 2$). The columns of E, S and D are filled with the data for the minimization problem, with columns concatenated as needed for the constraints. Indirect indexing will be used to tell the program which columns of the data matrices to use.

edex,sdex,ddex: Indirect address matrices for E, S and D matrices. The first *row* corresponds to the objective, and each subsequent *row* corresponds to a constraint. For example, E(:,edex(1,1:nr(1)*nt(1))) is the E data for the minimization criterion, and S(:,sdex(3,1:ns*nt(3))) is the S data for the second constraint. The data in S(:,sdex(3,1:ns*nt(3))) describes an (ns x nt(3)) impulse-response matrix.

con_type: Constraint type. A column vector of constraint (or minimization) type (1 for H_∞ and 3 for H_2). The total length is the number of rows in `edex`, `sdex`, and `ddex`.

gamma_hi: A vector of H_∞ constraint levels, corresponding to the γ_j in (1).

gamma_h2: A vector of H_2 constraint levels, corresponding to the γ_j in (1).

nf: The number of frequencies used to evaluate the transfer functions. This must be at least as large as the next power of 2 to the length of E.

nt: A vector of the number of disturbance inputs. The length matches the length of con_type.

ns: The number of control sensors (scalar).

nd: The number of control actuators (scalar).

nr: A vector of the number of regulated outputs. The length matches the length of con_type.

2.2 Initial Feasible Point

t0,W0: Starting values of W and t. They must be chosen so that <u>all</u> the constraints are feasible. For rules of thumb and a discussion on how to choose these starting values, please see the section "Finding a Feasible Starting Point." The dimension of `W0` is $W_{len} \times (ns * nd)$.

Z_hi0: Starting value for the Z associated with the H_∞ minimization and/or constraints. Its dimension is $(1+n_f/2) \times \sum(nt+nr)^2$ where n_f is the next power of 2 larger than the length of the `E` data. Compressed data format for Z0 is $diag\_cat((1/n_1 * eye(n_1)), eye(n_2), ...)$, where n_i is the dimension of the F matrix block for the i^{th} constraint.

a_z,N_z,A_z: Starting components for the Z associated with the H_2 minimization and/or constraints, where Z can be reconstructed by $Z = a_z I + A_z N_z (A_z)^T$. For calls to Mincode_hih2 which are *not* part of Phase I (finding a feasible starting point), these variables are set as follows: (Let i_{h2} be the indices of con_type equal to 3, signifying an H_2 constraint, and $ncon = length(con\_type(i_{h2}))$) Then $a\_z = [1/(le * nr(1) * nt(1) + 1); ones(ncon - 1, 1)]$; $N\_z = zeros(3 * ncon, 3)$; $A\_z = zeros(le * sum(nt(i_{h2}).*nr(i_{h2})) + ncon, 3)$.

2.3 Algorithm Knobs

Mt: A scalar giving an auxiliary constraint level on t which is the minimization variable in (2). `Mt` should be larger than any intermediate or final value of t.

MW: Auxiliary constraint levels on W. This matrix of dimension $W_{len} \times (ns * nd)$ should be larger than any intermediate or final value of W.

nu: A scalar signifying the weighting on duality gap versus nearness to analytical center. `Nu` must be greater than 1, and is generally set to 5.

tol_nd: A tolerance parameter signifying when to quit MINCODE. Once the duality gap is less than `tol_nd`, MINCODE exits. We recommend setting `tol_nd` equal to .001.

tol_ps: A tolerance parameter for the plane search. We recommend setting `tol_ps` equal to $1e{-}8$.

maxit_nd: The maximum number of Newton direction iterations allowed in MINCODE. At each iteration the duality gap decreases, and when it becomes smaller than `tol_nd`, Mincode_hih2 exits. Generally `maxit_nd` is set to 50.

maxit_ps: The maximum number of plane-search iterations. Generally this is set to a fairly large number, like 10,000.

3 Output Variables

t: The value of the objective function after minimization. That is, the value of $\|\mathbf{E}_0 + \mathbf{D}_0 \circledast \mathbf{W} \circledast \mathbf{S}_0\|_{p_0}$ at the solution.

W: The optimal value of the filter weights to achieve the minimization. W is a matrix of dimension $lw \times (ns * nd)$.

Z_hi: Information for piecing together the dual solutions for the H_∞ portions of the control problem.

a_z,N_z,A_z: Information for piecing together the dual solutions for the H_2 portions of the control problem (see initial conditions for these variables).

alpha: information about the dual feasibility error. Signifies how far off from dual feasibility we would be without the auxillary variables.

infor: Information regarding the exit conditions from MINCODE. This string eqither equals "ok", or else "maxiters exceeded" if MINCODE does not reach a duality gap smaller than `tol_nd` within `maxit_nd` number of iterations.

history: An array containing the history of the MINCODE iterations. Each row corresponds to an iteration, and contains iteration number, duality gap, alpha, step in primal direction, step in dual direction, and number of plane search iterations.

4 Finding A Feasible Starting Point

Caveat: Using Mincode_hih2 for a Phase I problem has not been fully tested.

To start the algorithm, a strictly feasible primal and dual point must be given ("strictly feasible" means that strict inequality holds in all the constraints). If all the constraints have E_j set to zero, (i.e. $\mathbf{E}_j = 0, j = 1, ..., k$ in (1)) then setting the initial primal variable $W0$ to all zeros, and $t0 > \|\mathbf{E}_0\|^2_{p_0}$ gives a strictly feasible primal point. A strictly feasible dual point is described in the description of the input variables `a_z`, `N_z`, `A_z`. However, if any of the $\mathbf{E}_j$ are nonzero and the constraint is less than $\|\mathbf{E}_j\|_{p_j}$ then a preliminary step is required. This procedure is referred to as a Phase I problem.

In a Phase I problem where there is just one constraint, use Mincode_hih2 to search for the solution to

$$\min \|\mathbf{E}_1 + \mathbf{D}_1 \circledast \mathbf{W} \circledast \mathbf{S}_1\|_{p_1}$$

and stop as soon as a $\mathbf{W}$ is found such that

$$\|\mathbf{H}_1\|_{p_1} \leq \gamma_1.$$

This $\mathbf{W}$ is the primal starting variable, and the resulting `Z_hi`, `a_z`, `N_z`, and `A_z` that are output from Mincode_hih2 are used as the initial values for calling Mincode_hih2 again on the full problem.

If there is more than one such constraint with a nonzero E_j, (e.g. the fourth constraint), then the $\mathbf{W}$ just found is used as a starting value in the new constrained problem

$$\begin{array}{ll} \text{minimize} & \|\mathbf{E}_4 + \mathbf{D}_4 \circledast \mathbf{W} \circledast \mathbf{S}_4\|_{p_4} \\ \text{subject to} & \|\mathbf{E}_1 + \mathbf{D}_1 \circledast \mathbf{W} \circledast \mathbf{S}_1\|_{p_1} \leq \gamma_1. \end{array}$$

The algorithm can be stopped as soon as a new $\mathbf{W}$ is found such that $\|\mathbf{E}_4 + \mathbf{D}_4 \circledast \mathbf{W} \circledast \mathbf{S}_4\|_{p_4} \leq \gamma_4$. Otherwise, if the solution to this constrained problem is found and it does not satisfy $\|\mathbf{E}_4 + \mathbf{D}_4 \circledast \mathbf{W} \circledast \mathbf{S}_4\|_{p_4} \leq \gamma_4$ then no feasible point exists. This procedure of calling Mincode_hih2 to solve increasingly larger constrained minimization problems continues until all constraints have been accounted for.

TO: Users of MINCODE
FROM: Tom Nixon
DATE: March 18, 1996
SUBJECT: *New MINCODE Driver and Data Builders*

1 Introduction

This memo is to describe how to use a new MINCODE driver and it's associated data building routines. These routines provide a simple to use interface to the low level MINCODE engine and provide tools to easily build the necessary input data structures.

Section 2 of this memo discusses in detail the new MINCODE driver and the associated data building functions. A quick overview of each function exists in Appendix I. Section 3 gives a complete example problem using the data building routines and the MINCODE driver.

All of the functions mentioned in this memo exist in /home/updike/usr3/pauljt/5796/mincode/LatestMincode/TAR. I encourage you to use them, as feedback would be greatly appreciated.

2.1 The MINCODE data

As you may already be familiar, the MINCODE engine is called in the following manner:

```
[t, W, Z_hi, a_z, N_z, A_z, alpha, infor, history] = ...
    Mincode_hih2(E, edex, S, sdex, D, ddex, con_type, ...
         gamma_hi, gamma_h2, nf, nt, ns, nd, nr, ...
         t0, W0, Z_hi0, a_z0, N_z0, A_z0, ...
         Mt, MW, nu, tol_nd, tol_ps, maxit_nd, maxit_ps);
```

For a detailed description of these input variables, refer to the technical memo [1].

The input variables may be broken down into three groups:

1. Specifications of Problem and Data: E, edex, S, sdex, D, ddex, con_type, gamma_hi, gamma_h2, nf, nt, ns, nd, and nr.

2. Initial Feasible Point: t0, W0, Z_hi0, a_z0, N_z0, and A_z0.

3. Algorithm knobs: Mt, MW, nu, tol_nd, tol_ps, maxit_nd, and maxit_ps.

To keep to a minimum the number of input data elements that the user must keep track of, a single structure for each group has been created. These structures are called: *Data, Inits, and Knobs*. respectively. All of the information from the variables in each group exists in one single variable. Routines have been provided to easily create these new structures.

2.2 The *Data* structure and associated routines

Data is built by calling the ***BuildData()*** routine once for the minimization problem and then once for each constraint needed. The *Data* structure grows with each successive call to the ***BuildData()*** routine. ***BuildData()*** is called in the following manner:

When describing the minimization problem:

```
data = BuildData(data, E, D, S, con_type, ...
     ns, nd, nt, nr, nf);
```

Note that in the first call to ***BuildData()*** *data* should be empty ([]). The variables con_type, nt, and nr are the scalar values used only in the minimization problem.

When adding a constraint:

```
data = BuildData(data, E, D, S, con_type, gamma);
```

Note that the variable *data* used as input to this call of ***BuildData()*** is the variable *data* returned from the previous call (*data* may not be empty when adding a constraint). The variables ns, nd, and nf must remain the same for the minimization problem and the constraints, so they are not needed as input when adding a constraint. The variables con_type and gamma are unique to each constraint and must be supplied each time a new constraint is added. The variables nt and nr may be different for each constraint added, but need not be supplied as they will be calculated from the sizes of E, D, and S.

Frequently, users may want to supply the 0 matrix or the Identity Matrix Transfer Function, in place of E, D, or S. Instead of unnecessarily creating these matrices to be used in the call to ***BuildData(),*** users may simply enter '0', 0, 'I', or 'α I' (I multiplied by a constant) as a shortcut. Another frequently used matrix is the following:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & . & . & . & \alpha_n \\ 0 & 0 & . & . & . & 0 \\ . & . & . & & & \\ . & . & & . & & \\ . & . & & & . & \\ 0 & 0 & & & & 0 \end{bmatrix}$$

This is a matrix composed of vectors whose first element is a scalar and each subsequent element is 0. Matrices such as these can be passed to ***BuildData()*** by entering the string '$\alpha_1, \alpha_2, \ldots, \alpha_n$'. Note that there must be as many α's as there would be columns for that matrix. As this is a literal string, the actual floating point numbers must be entered, not variable names.

For an example, suppose you are trying to solve the following problem:

$$
\begin{array}{lll}
\textit{min } W & \| E_0 + D_0 \otimes W \otimes S_0 \|_\infty & \\
\textit{subject to} & \| W \otimes S_1 \|_2 \leq \gamma & \text{Actuator Impulse Energy Constraint} \\
 & \| \Delta C \otimes W \|_\infty < 1 & \text{Stability Robustness Constraint}
\end{array}
$$

The **MATLAB** calling sequence to build *data* would look like this:

```
% Remember to use con_type = 1 for Hinf
%                 and con_type = 3 for H2

data = BuildData([], E0, D0, S0, 1, ns, nd, nt, nr, nf);
data = BuildData(data, '0', 'I', S1, 3, gamma);
data = BuildData(data, '0', Delta_C, 'I', 1, 1);
```

You may associate any valid **MATLAB** string as a comment with the *data* matrix by calling:

```
str = ['Constraint 1: Actuator Impulse Energy. ' ...
         'Constraint 2: Stability Robustness.'];
data = AddComment(data, str);
```

You may retrieve the associated comment from a *data* matrix by calling:

```
comment = GetDataComment(data);
```

If you wish to see the actual matrices (E, edex, etc.) generated, you may obtain this information by making a call to *UnpackData()* which is called in the following manner:

```
[E, D, S, edex, ddex, sdex, ns, nd, nt, nr, nf,   ...
     con_type, gamma_hi, gamma_h2, comment] = ...
     UnpackData(data);
```

If you already have the individual input matrices from a previous problem that you would like to pack into the new *data* matrix structure, you may call *PackData()* in the following manner:

```
data = PackData(E, D, S, edex, ddex, sdex, ...
     ns, nd, nt, nr, nf, ...
     con_type, gamma_hi, gamma_h2, comment);
```

To retrieve the data matrices associated with any one (1) constraint or the minimization problem use *GetData()*. The minimization information is considered element 0 and the constraints make up elements 1 through the number of constraints.

```
num = 2; % Retrieving data from the second constraint.
[E, D, S, ns, nd, nt, nr, nf, con_type, gamma] = ...
     GetData(data, num);
```

To delete a constraint, use *DelDataCon()*:

```
constraint_num = 1;    % Deleting the first constraint.
                         constraint_num may not equal 0, as
```

```
                                    you are not allowed to delete the
                                    minimization problem's
                                    information.
data  =  DelDataCon(data,  constraint_num);
```

Let's say that in the example problem above, we choose to change gamma in the first constraint to gamma_new. Four examples are given for how this could be performed.

Example 1: Start over completely.

```
data = BuildData([], E0, D0, S0, 1, ns, nd, nt, nr, nf);
data = BuildData(data, '0', 'I', S1, 3, gamma_new);
data = BuildData(data, '0', Delta_C, 'I', 1, 1);
```

Example 2: Use *UnpackData()* & *PackData()*.

```
[E, D, S, edex, ddex, sdex, ns, nd, nt, nr, nf,   ...
     con_type, gamma_hi, gamma_h2, comment] = ...
     UnpackData(data);
gamma_h2(1) = gamma_new;
data = PackData(E, D, S, edex, ddex, sdex, ...
     ns, nd, nt, nr, nf, ...
     con_type, gamma_hi, gamma_h2, comment);
```

Example 3: Use *DelDataCon()* & *BuildData()*.
(Note that this example will reorder the constraints)

```
data = DelDataCon(data, 2);
data = BuildData(data, '0', 'I', S1, 3, gamma_new);
```

Example 4: Use *GetData()*, *DelDataCon()*, & *BuildData()*.
(Note that this example will reorder the constraints)

```
[E, D, S, ns, nd, nt, nr, nf, con_type, gamma] = ...
     GetData(data, 2);
data = DelDataCon(data, 2);
data = BuildData(data, E, D, S, con_type, gamma_new);
```

2.3 The *Inits* structure and associated routines

The *Inits* variable which contains information for t0, W0, Z_hi0, a_z0, N_z0, and A_z0 may be packed into a single variable with the ***PackInits()*** function. All the individual matrices may be retrieved from the single *inits* variable with the ***UnpackInits()*** function. These routines are called in the following manner:

```
inits = PackInits(t0, W0, Z_hi0, a_z0, N_z0, ...
     A_z0, comment);

[t0, W0, Z_hi0, a_z0, N_z0, A_z0, comment] = ...
     UnpackInits(inits);
```

The function, *PickStartingInits()* , may be used to create default starting values for the *inits* variable. See [1] for a detailed explanation of how these defaults are generated.

PickStartingInits() takes as input an already filled *data* variable and optionally, the length of the desired weights (*lw*). If *lw* is not supplied, *PickStartingInits()* will choose an appropriate length based on the information in *data*. The MATLAB call looks like the following:

```
inits = PickStartingInits(data, lw);
```

Called as above, the routine will create a W0 matrix of all 0's. This unfortunately is not always a feasible starting point for W0. Similarly, t0, and the dual variables (*Z_hi0, a_z0, N_z0,* and *A_z0*) generated by *PickStartingInits()* may also not represent a feasible starting point. Users may instead have to create these variables through some other method (maybe Phase I runs of MINCODE). They may then use the *PackInits()* function to place their matrices into the *inits* variable.

2.4 The *Knobs* structure and associated routines

The *Knobs* variable which contains information for Mt, MW, nu, tol_nd, tol_ps, maxit_nd, and maxit_ps may also be packed into a single variable with the *PackKnobs()* function. All the individual matrices may be retrieved from the single *knobs* variable with the *UnpackKnobs()* function. These routines are called in the following manner:

```
[Mt, MW, nu, tol_nd, tol_ps, maxit_nd, maxit_ps, ...
     comment] = UnpackKnobs(knobs);

knobs = PackKnobs(Mt, MW, nu, tol_nd, tol_ps, ...
     maxit_nd, maxit_ps, comment);
```

The function, *PickStartingKnobs()*, may be used to create default starting values for the *knobs* variable. See [1] for a detailed explanation of how these defaults are generated. *PickStartingKnobs()* takes as input the already filled *data* and *inits* variables. The MATLAB call looks like the following:

```
knobs = PickStartingInits(data, inits);
```

If the default *knobs* values are not sufficient for any particular problem, users may generate their own values and then fill the *knobs* variable with the *PackKnobs()* function.

2.5 The MINCODE driver

The new MINCODE driver is designed both to take a variable number of inputs and to generate a variable number of outputs depending on how the user intends to call the function. The most general calling sequence looks like the following:

```
[t, W, Z_hi, a_z, N_z, A_z] = ...
     MincodeDriver(data, inits, knobs);
```

Notice first that the output matrices are the exact same matrices that make up an *inits* data structure. If the number of output arguments is either 0 or 1, *MincodeDriver()* assumes that the user would prefer to have all of the outputs packed into the *inits* data structure for them. This would change the calling sequence to:

```
results = MincodeDriver(data, inits, knobs);
```

Retrieving the Weights (W) or any other information from *results* can be done with *UnpackInits()*. If the number of output arguments is greater than 1, *MincodeDriver()* will not pack the results into one variable, but instead will return only the number of outputs requested. For instance, if only t and W are wanted as output, the user may call *MincodeDriver()* like this:

```
[t, W] = MincodeDriver(data, inits, knobs);
```

The only mandatory input argument is *data*. If either *inits* and/or *knobs* are not supplied, *MincodeDriver()* will use the default values obtained from *PickStartingInits()* and *PickStartingKnobs()*. If the user chooses not to supply *inits*, they must instead give a value for *lw* so the length of the default *W0* can be set. This leads to any of the following calling sequences:

```
results = MincodeDriver(data, lw);
results = MincodeDriver(data, lw, knobs);
[t, W, Z_hi] = MincodeDriver(data, lw);
```

3 A complete example problem

Submerged Spherical Shell: The plant has two disturbances, two control sensors, two control actuators, and two radiated-field sensors. We will reduce the disturbances' acoustic radiation while simultaneously constraining a stability robustness measure, performance sensitivity, actuator authority, and out-of-band enhancement. See [2] for more details of sample problem.

High-level specifications on the controlled system are as follows:

1. Reduce acoustic radiation over the band 40-180 Hz.
2. Guarantee closed-loop stability for a specified level of output-multiplicative uncertainty in the coupling C.
3. Constrain (to α_2) the closed-loop transfer function's sensitivity to additive uncertainty in the actuator response D.
4. Constrain (to α_3) the closed-loop transfer function's sensitivity to additive uncertainty in the sensor response S.
5. Limit (to α_4) the RMS response of actuator 1 to unit-variance white noise disturbance inputs.
6. Limit (to α_5) the RMS response of actuator 2 to unit-variance white noise disturbance inputs.
7. Limit (to α_6) the white-noise-to-RMS out-of-band response.

Mathematical formulation:

$$
\begin{aligned}
\textit{min } W \quad & \| E_0 + D_0 \otimes W \otimes S_0 \|_2 \\
\textit{subject to} \quad & \| W \otimes C \|_\infty \le \alpha_1 \\
& \| W \otimes S_2 \|_\infty \le \alpha_2 \\
& \| D_0 \otimes W \otimes \beta I \|_\infty \le \alpha_3 \\
& \| (U1,0) \otimes W \otimes S_2 \|_2 \le \alpha_4 \\
& \| (0,U1) \otimes W \otimes S_2 \|_2 \le \alpha_5 \\
& \| E_6 + D_0 \otimes W \otimes S_6 \|_2 \le \alpha_6
\end{aligned}
$$

Scenario:

1. Solve the basic problem.

```
% Put the MINCODE engine and MINCODE driver in your path.

path(path, '/home/updike/usr3/pauljt/5796/mincode/LatestMincode/TAR');

% Build the data structure

Hinf = 1;
H2 = 3;
data = BuildData([], E0, D0, S0, H2, ns, nd, nt, nr, nf);
data = BuildData(data, 0, 'I', C, Hinf, alpha(1));
data = BuildData(data, 0, 'I', S2, Hinf, alpha(2));
data = BuildData(data, 0, D0, sprintf('%.15e I', Beta), Hinf, alpha(3));
data = BuildData(data, 0, '1,0', S2, H2, alpha(4));
data = BuildData(data, 0, '0,1', S2, H2, alpha(5));
data = BuildData(data, E6, D0, S6, H2, alpha(6));

% Solve the problem

results = MincodeDriver(data, lw);

% Extract the Weights (W) from results and solve for performance.

[t, W, Z_hi, a_z, N_z, A_z, comment] = UnpackInits(results);
Performance(data, W);     % This function is not yet available.
```

2. Do a tradeoff study of performance versus multiplicative uncertainty allowed in C while maintaining closed-loop stability. That is change the α_1 value several times and solve for the performance, creating a trade-off curve.

```
[E, D, S, edex, ddex, sdex, ns, nd, nt, nr, nf, ...
        con_type, gamma_hi, hamma_h2, comment] = UnpackData(data);
for i = 1:num_alpha1s,
```

```
        gamma_hi(1) = alpha1(i);
        data = PackData(E, D, S, edex, ddex, sdex, ns, nd, nt, nr, nf, ...
                con_type, gamma_hi, gamma_h2, comment);
        [t, W] = MincodeDriver(data, lw);
        Performance(data, W);     % Again, this function is not yet available.
end
```

3. Go back to the original problem, but delete the second constraint, ignoring any additive uncertainty in D.

```
data = DelDataCon(data, 2);
[t, W] = MincodeDriver(data, lw);
Performance(data, W);     % Again, this function is not yet available.
```

Appendix I: MINCODE Driver functions

The *Data* Structure Routines

AddComment Appends a valid MATLAB string to a *Data* variable.

```
data = AddComment(data, str);
```

BuildData Adds information about the minimization problem or the constraints to a *Data* variable.

```
% For the minimization problem
data = BuildData([], E, D, S, con_type, ns, nd, nt, nr, nf);

% For constraints
data = BuildData(data, E, D, S, con_type, gamma);
```

DeleteDataCon Deletes any one constraint from a *Data* variable.

```
data = DelDataCon(data, constraint_num);
```

GetData Retrieves specific matrices associated with the minimization problem or any one constraint.

```
[E, D, S, ns, nd, nt, nr, nf, con_type, gamma] = GetData(data, level);
```

GetDataComment Retrieves a comment string from a *Data* variable.

```
str = GetDataComment(data);
```

PackData Packs all the matrices associated with the specification of the problem and the constraints into a single variable.

```
data = PackData(E, D, S, edex, ddex, sdex, ns, nd, nt, nr, nf, ...
        con_type, gamma_hi, gamma_h2, comment);
```

UnpackData Retrieves all the matrices associated with the specification of the problem and the constraints from a *Data* variable.

```
[E, D, S, edex, ddex, sdex, ns, nd, nt, nr, nf, con_type, gamma_hi, ...
        gamma_h2, comment] = UnpackData(data);
```

The *Inits* Structure Routines

PackInits Packs all the information associated with the initial feasible point into a single variable.

```
inits = PackInits(t0, W0, Z_hi0, a_z0, N_z0, A_z0, comment);
```

PickStartingInits Picks Default starting values for the *inits* variable. These defaults may not always describe a feasible starting point.

```
inits = PickStartingInits(data, lw);
```

UnpackInits Retrieves all the information associated with the initial feasible point from a single *Inits* variable.

```
[t0, W0, Z_hi0, a_z0, N_z0, A_z0, comment] = UnpackInits(inits);
```

The *Knobs* Structure Routines

PackKnobs Packs all the information associated with the algorithm knobs into a single variable.

```
knobs = PackKnobs(Mt, MW, nu, tol_nd, tol_ps, maxit_nd, maxit_ps,
comment);
```

PickStartingKnobs Picks Default starting values for the *knobs* variable. These defaults may not be suitable for every problem.

```
knobs = PickStartingKnobs(data, inits);
```

UnpackKnobs Retrieves all the information associated with the algorithm knobs from a single *Knobs* variable.

```
[Mt, MW, nu, tol_nd, tol_ps, maxit_nd, maxit_ps, comment] = ...
        UnpackKnobs(knobs);
```

The MINCODE Driver

MincodeDriver Hi level routine used to solve a **MINCODE** problem. This routine calls the low level MINCODE engine function, **Mincode_hih2.**

```
[t, W, Z_hi, a_z, N_z, A_z] = MincodeDriver(data, inits, knobs);
```

References

[1] Julia Olkin and Paul J. Titterton, Jr. *"Description of Input/Output Variables for MINCODE"*, Internal SRI technical memo. December 20, 1995.

[2] Paul J. Titterton, Jr. and Julia A. Olkin, "A practical method for constrained-optimization controller design: H2 or H∞ optimization with multiple H2 and /or H∞ constraints." To appear in *Proceedings of the (IEEEs) 29th Asilomar Conference on Signals, Systems and Computers, Oct. 30-Nov. 1, 1995.*

SRI International

Technical Note • April 1996

PERFORMANCE ROBUSTNESS USING MINCODE

Prepared by:

PAUL J. TITTERTON, *Research Physicist*
JULIA A. OLKIN, *Research Engineer*
WILLIAM C. NOWLIN, *Senior Research Engineer*
Applied Control and Signal Processing Group
System Technology Division

Prepared for:

Office of Naval Research
Code 1222
800 N. Quincy St.
Arlington, VA 22217

Attn: Dr. Geoffrey L. Main
Program Manager, Structures

Contract N00014-94-C-0128

SRI Project 5796

A technical note is a working paper that presents the results of research related to a single phase or facet of a research problem. It presents the concepts, findings, and/or conclusions of the authors, alone, in fragmentary form intended for internal discussion and comment. Such notes are subject to revision, incorporation into more formal reports, or withdrawal.

Approved by:

J. RAUL MARTINEZ, *Director*
Applied Control and Signal Processing Group

PERFORMANCE ROBUSTNESS USING MINCODE

1 SUMMARY

This report describes initial results in synthesizing controllers that meet *performance robustness* constraints. To achieve performance robustness, we require that a control design maintain disturbance-to-regulated-output dynamics at a prescribed level over some particular ensemble of plants describing an operational envelope. Our goal is to synthesize a design directly based on performance robustness specifications.

For any given control law design, the critical quality of performance robustness is usually analyzed after the fact, with at best an indirect set of steps to follow if the robustness is found inadequate. Here we demonstrate a controller design for reducing vibration from a spherical shell such that the performance level is maintained for a set of perturbations to the measured coupling. The synthesis is based on new constrained optimization techniques, uses the MINCODE software package, proceeds directly without cumbersome iterations, and is unique in the field. In future efforts, we plan to extend this robust performance methodology to the synthesis of MIMO PSI controllers.

In conclusion we have

- Solved performance robustness problem for SISO disturbance rejection controllers
- Synthesized performance-robust controllers *directly from specifications*, using convex optimization and MINCODE
- Eliminated suboptimal "cut and try" design techniques representing former state of the art
- Verified results on test problem

2 PERFORMANCE ROBUSTNESS

We design a controller for the SISO disturbance-rejection configuration shown in figure 1. The Δ represents the uncertainty to the system, and we assume that Δ is linear-time-invariant and bounded, that is, $\|\Delta\|_\infty < 1$. The variability is frequency-shaped by the transfer function G, which enables us to model some parts of the frequency band as having more variability than other parts.

Let Q denote the FIR discretization of the Q-parameter. The nominal performance condition is given by

$$\|BS\| < 1; \qquad \text{where} \qquad S = 1 - QC. \tag{1}$$

We refer the reader to figure 2 for an aid to the following explanation. Suppose the disturbance input to the system has a power spectral density equal to that of the performance weighting curve, B. Then the performance criterion is recognized as a whitening criterion that tries to reduce the disturbance in the control band to a prescribed level, while allowing some enhancement out of band, so that the resulting output is statistically white.

The robust stability condition is given by

$$\|GT\| < 1; \qquad \text{where} \qquad T = QC. \tag{2}$$

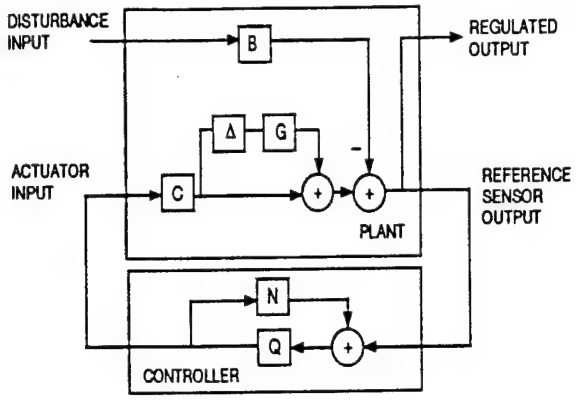

Figure 1: SISO disturbance-rejection configuration. The blocks are a simplification of our standard E, S, D, C representation. B is a performance weighting transfer function; C is the nominal coupling, and multiplicative variability in C is represented by the two blocks, Δ and G. G is a constant variability weighting matrix, and Δ is unknown but bounded. The controller is represented as a Q-parameterization.

This is the normal stability criterion that we have employed often.

We now compute, and subsequently bound, the effect of the changing Δ on performance. One can show that the perturbed sensitivity becomes

$$S_{perturb} = \frac{S}{1 + \Delta GT}.$$

The robust performance conditions are thus

$$\|GT\|_\infty < 1 \quad \text{and} \quad \|BS_{perturb}\|_\infty < 1 \text{ for every } \Delta \text{ such that } \|\Delta\|_\infty < 1. \tag{3}$$

Doyle *et al.*[1] showed that a necessary and sufficient condition that equation 3 is satisfied is

$$\| |BS| + |GT| \|_\infty < 1. \tag{4}$$

Robust performance of a controller can be analysized by simply evaluating the expression in (4). However, a robustly performing controller can also be *synthesized* because the expression in (4) is the sum of two affine functions in Q, and is thus convex and can be constrained using MINCODE methods. This approach is new, and its implementation extends the state of the art for solving performance robustness problems. This methodology, if successfully extended to MIMO designs, would apply directly to synthesis of the CRADLE controller for disturbance rejection.

3 SAMPLE PROBLEM

To demonstrate this new methodology, we have prepared an example using a simple submerged spherical shell. The plant has one input and one output taken from the description given in our

[1]Doyle, Francis, and Tannenbaum, "Feedback Control Theory", MacMillan 1992.

Asilomar paper.[2] Here we use (4) to synthesize a performance-robust controller and test it against synthesized variability. The filter G is a highpass filter for variability weighting, and the filter B is a lowpass filter for performance weighting. We chose B to reflect a disturbance rejection problem where the disturbance spectral content is shaped like B, so that the system output would be flat with at most unit output. Both filters are shown in figure 2, plotted against normalized angular frequency (from DC to 1=Nyquist). The goal, then, is to design a controller so that any perturbed system also produces a flat output with a level no larger than unity.

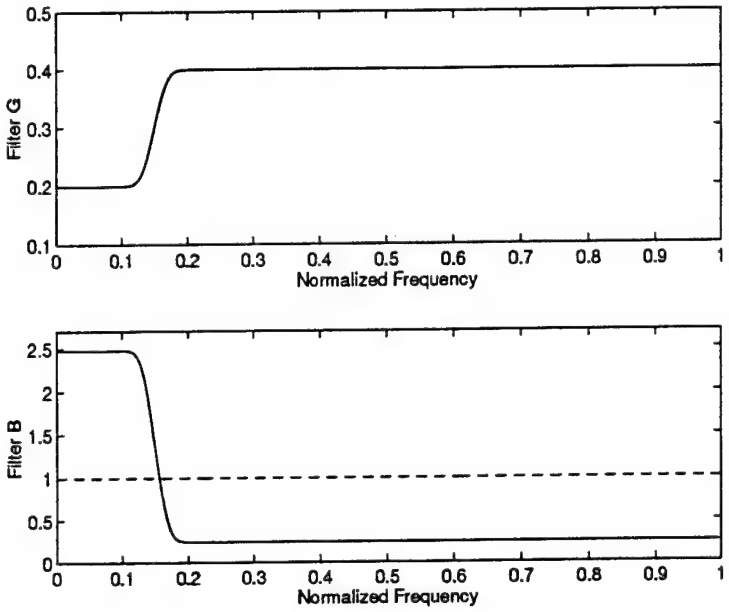

Figure 2: Variability weighting filter G and performance weighting filter B. The B filter can be considered to mimic the spectral content of the disturbance signal for a whitening disturbance rejector. The horizontal line with unit amplitude corresponds to the maximum allowable closed-loop response for any plant variation in the ensemble, as a function of frequency.

3.1 Controller Synthesis

We use MINCODE to solve the following constrained optimization problem for the optimal controller Q:

$$\begin{array}{ll} \text{minimize} & \|BS\|_\infty = \|B - BQC\|_\infty \\ \text{subject to} & \|GT\|_\infty = \|GQC\|_\infty < \beta. \end{array} \tag{5}$$

Notice that $Q = 0$ is a feasible point and thus can be used as a starting point for MINCODE.

For a particular value of β the optimization problem in (5) is solved to give the optimal Q. The condition in (4) is checked by substitution. If the condition is satisfied then we have designed a controller which maintains performance robustness.

We ran MINCODE on (5) for values of β ranging from .1 to 1.5. Each value of beta gives a corresponding $\|BS\|_\infty$ and $\|GT\|_\infty$. In figure 3 we show the curve described by the various values

[2]Paul Titterton, Jr. and Julia Olkin, "A practical method for constrained optimization controller design: H^2 or H^∞ optimization with multiple H^2 and/or H^∞ constraints" in the Proceedings of the 29th Asilomar Conference on Signals, Systems and Computers, 1995.

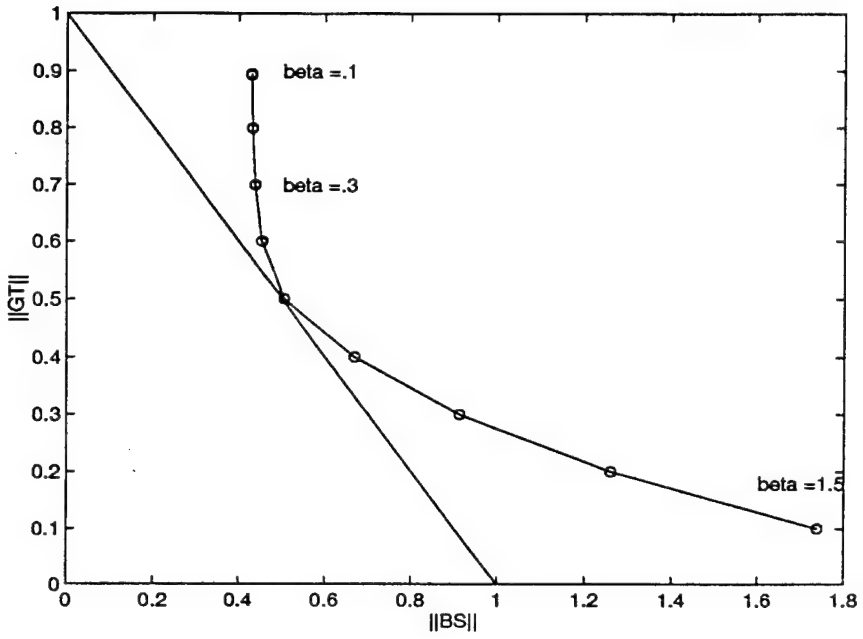

Figure 3: $\|BS\|_\infty$ versus $\|GT\|_\infty$ for values of $\beta = \{.1, .2, \ldots, 1.5\}$

of β, having first drawn the line segment where $\|BS\|_\infty + \|GT\|_\infty = 1$. (We caution the reader that this expression is an approximation to (4), but is by design a good one. Under mild conditions, which we meet in this example, the two expressions are equal.) All the values of β that yield points below the line segment generate controllers that meet the performance requirements. However, as we choose controllers corresponding to points on the curve closer to the origin, we lose some amount of performance. Thus, we try to choose a point on the β-parameterized curve that just touches the line segment. This represents a controller which meets the performance robustness criterion, but "just barely," while providing as much overall performance as possible.

3.2 Validation

To validate that the criteria for robust performance was met, a Monte Carlo simulation was performed. The controller used for the nominal system corresponds to the solution of (5) for $\beta = .5$. The perturbation set Δ is chosen randomly so that $\|\Delta\|_\infty < 1$, with the attempt to sample the full range of such bounded variations. The resulting robust performance conditions in (3) were verified. Hundreds of perturbations were considered, and as the scatter plot in figure 4 shows, the synthesized controller maintains the specified performance level. Each x mark corresponds to a perturbation, and the resulting norm $\|BS_{perturb}\|$ should be less than one, signifying that the performance specification has been satisfied. The robust performance region is the area in the figure to the left of the line segment at $\|BS_{perturb}\| = 1$, and all points to the right of the line would be in the nonrobust performance region.

Finally, we present a sample performance plot for a few variations, Δ. As figure 5 shows, performance remains better than the unit response across the entire band, despite the fact that the controller was designed without specific knowledge of the particular Δ used here.

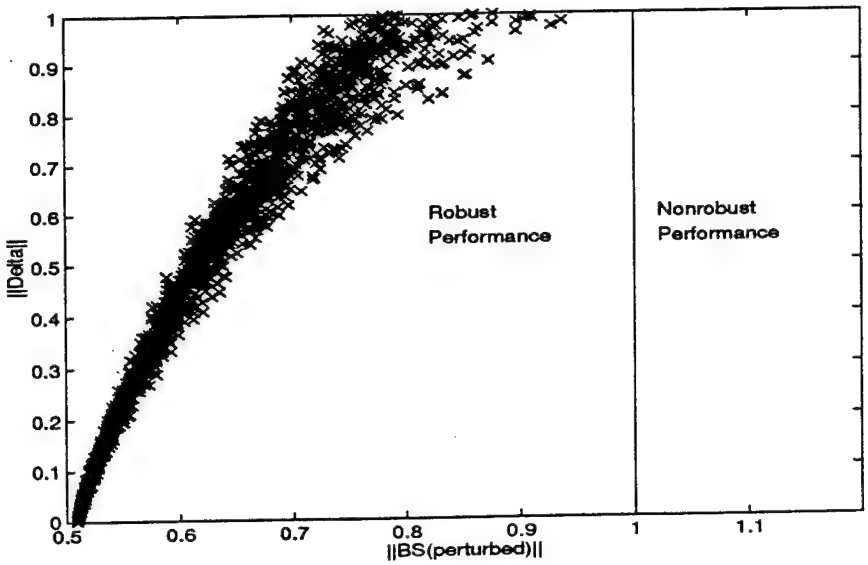

Figure 4: $\|BS_{perturb}\|$ for perturbations Δ. The robust performance region is the area to the left of the line segment at $\|BS_{perturb}\|$.

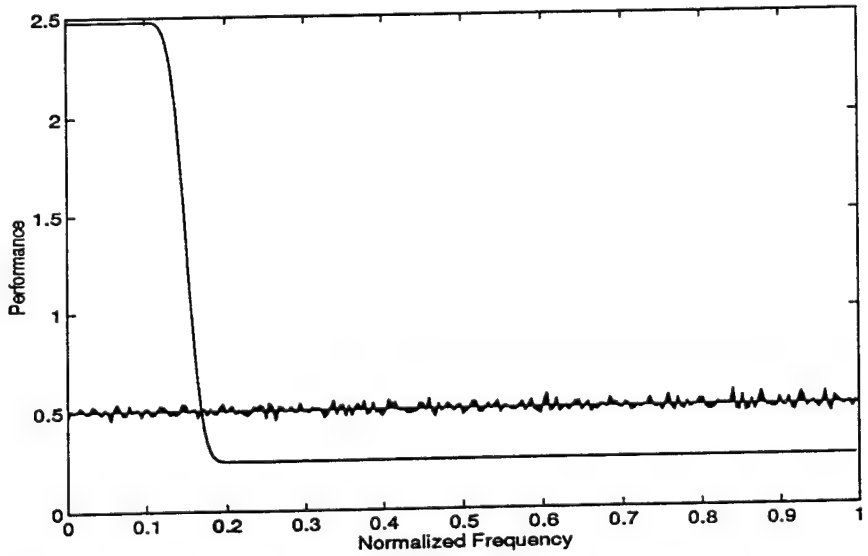

Figure 5: Performance for a few instances of Δ. Performance robustness is maintained in this check.

INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL, VOL. 6, 413–430 (1996)

PRACTICAL MULTI-CONSTRAINT H^∞ CONTROLLER SYNTHESIS FROM TIME-DOMAIN DATA

PAUL J. TITTERTON, JR.*

System Technology Division, SRI International, 333 Ravenswood Ave., Menlo Park, California 94025, U.S.A.

SUMMARY

This paper presents a practical algorithm for MIMO controller design with multiple H^∞ norm constraints. The plant is described by its sampled-data impulse response matrix, which could be determined directly from measurements; the controller design parameters are the tap weights of an FIR discretization of the Q- (or Youla-) parameter. We approximate the H^∞ norm by the maximum transfer matrix 2-norm on an even frequency sampling over $\theta = [0, \pi]$. The control design algorithm uses the Newton method to minimize two cost functions sequentially: the first determines multiple H^∞ feasibility, and the second minimizes a generalized entropy. As a design example, we control the acoustic radiation from a (mathematically modelled) submerged spherical shell. The plant-model impulse response matrix has McMillan degree 8 800. We specify and synthesize a controller that simultaneously achieves ten H^∞ constraints: one on the radiation reduction, one for stability robustness, and eight on actuator authority.

KEY WORDS H^∞-norm controller design; constrained optimization; interior-point methods

1. INTRODUCTION

This paper describes a practical algorithm for linear controller design with multiple H^∞-norm constraints. We assume that the plant is well represented by a sampled-data finite-length impulse response (FIR) matrix. Because of this plant representation, the algorithm described here is appropriate for control of infinite- as well as finite-dimensional systems. We also provide a design example: multi-sensor, multi-actuator control of the the acoustic radiation from a submerged spherical shell. We impose a small-gain stability robustness constraint and individual actuator authority constraints. This example demonstrates the algorithm's utility for controller design on realistically large, dynamically complex plants.

The standard discrete-time H^∞ norm feasibility criterion is written:

$$\sup_{\theta} \bar{\sigma}(\mathbf{H}(e^{j\theta})) < \gamma \tag{1}$$

where $\mathbf{H}$ is a stable transfer function, θ is the normalized angular frequency, $\bar{\sigma}$ is the largest singular value of the matrix argument and γ is the given bound. We approximate this criterion

* Also at the Space, Telecommunications and Radioscience Laboratory, Stanford University, Stanford, California 94305, U.S.A.

CCC 1049–8923/96/050413–18
© 1996 by John Wiley & Sons, Ltd.

by

$$\max_k \bar{\sigma}(\mathbf{H}(e^{j\theta_k})) < \gamma \quad \text{for } k = 1 \text{ to } N_f \tag{2}$$

where the θ_k are the non-negative discrete Fourier transform (DFT) bin frequencies associated with the FIR plant. This is a natural approximation for an FIR description of $\mathbf{H}$, which can be zero-padded to produce an arbitrarily fine sampling of the half unit circle. For the remainder of this paper, we refer to Equation (2) as the H^∞ norm constraint, and we take $\mathbf{H}$ to be some closed-loop transfer function associated with the controlled plant.

Enforcing multiple H^∞ norm constraints greatly increases the utility of H^∞ concepts in practical applications.[1,2,3] The current H^∞ controller design method of choice—the coupled Ricatti equation method[4]—requires heuristic and iterative weighting matrix modification to achieve desired constraints. Also, it often leads to closed-loop plant characteristics that are little different from those achievable with H^2 design.[5] Specifications on the stability robustness and actuator authority, say, cannot be individually enforced. The method presented in this paper allows us to specify, and simultaneously satisfy, several H^∞ constraints. Of course, weighting matrices can be used to shape the frequency response and emphasize input and output combinations for any of the constrained closed-loop transfer functions. Because the weighting matrices are applied constraint by constraint, their choice is often simple and direct.

Apart from the constraints, this controller design method has three main differences from the primary body of control literature: the plant model (and class of plants to which the algorithm may be practically applied), the controller parametrization, and the numerical algorithms.

We assume an FIR representation for the plant, and thus we restrict ourselves to stable plants with relatively short impulse responses. The limit on FIR length is due to computational resources. Some plants that are dynamically complex are well represented this way, and others that are dynamically simple have impulse responses impractically long—the radiation from a submerged spherical shell and the displacement of a lightly damped oscillator, respectively, are good examples. This method can be applied to unstable or long-impulse-response state-space plants by first designing a stabilizing observer-based controller[6,7] and then representing the well-damped plant as a matrix FIR[1].

This plant model, although naturally finite-dimensional, can be used to describe infinite-dimensional plants or plants with very large state-transition matrices. Perhaps most importantly, in cases such as vibration control of damped structures, this model can be taken directly from measured data with only minimal preprocessing. Thus the model's accuracy in practical application is ensured: there is no need for analytic model generation, state-space system identification or model reduction.

There are, however, plants and measurement apparatus which do not allow simple measurement of FIR models: for example, we may not have access to the disturbance. Additionally, the data may be particularly noisy, requiring extensive coherent averaging to attain a good model. For dynamically simple plants, state-space system identification procedures incorporate this averaging into estimation of their (relatively few) parameters. For the same plant, because of model parameter redundancy, the FIR estimation may require proportionally more measured data. Of course, the computational ease of FIR estimation can offset the redundancy and measurement requirements.

For controller design, we use the simplest Q- (or Youla-) parametrization possible for stable plants. As per Boyd and Barrett[2] (Section 7.3.1), a parallel and negated transmission path in the control hardware[7] neutralizes the control-actuator to control-sensor transfer function. The Q-parameter is realized as a filter from the output of the neutralized plant to the control-actuator

input. More complicated Q-parametrizations can be used (those for unstable or long-impulse-response systems), but they require an initial stabilizing controller design and implementation.

We follow Boyd *et al.*[1] and Peterson *et al.*[8] in using a matrix FIR representation for Q, where our design parameters are the individual tap weights. Peterson *et al.* use hardware[9,10] that implements a matrix FIR Q-parameter and a matrix FIR neutralizing path. As such, this hardware is suitable for implementing this paper's controllers.

Two previous design algorithms exist for this plant model and controller parametrization. Boyd *et al.*[1] developed a general control design compiler, **qdes**, around a linearly constrained quadratic optimization engine. Olkin *et al.*[11] developed a (single constraint) quadratically constrained quadratic optimization engine that exploits the problem structure. The latter code was used by Peterson *et al.*[8] and is used for this paper's preliminary H^2 design.

The algorithm developed here for multi-constraint H^∞ norm controller design is conceptually simple. The first step determines feasibility, and, if the problem proves feasible, the second step minimizes a generalized entropy. In both, we define a convex objective function (quasi-convex in the feasibility step) and use Newton's method to minimize that function. For infeasible problems, if exact Newton directions can be found, the first step has polynomial complexity. For feasible problems, the second step has polynomial complexity and a rigorous stopping criterion.[12,13]† The polynomial complexity and rigorous stopping criterion allow us to make definitive statements about feasibility and the minimum generalized entropy.

In most problems of practical interest, Q has thousands (to tens of thousands) of free parameters, and exact solution of the Newton direction is impractical or impossible. Without the exact solution, we lose polynomial complexity and are forced to use heuristic stopping criteria, with the implication that negative feasibility results are not definitive. Even without the exact Newton direction, this method has proven to be highly useful. This utility is due to the fact that, if the Q-parameter used to initialize the algorithm is feasible for some number of the constraints, then those constraints will remain feasible throughout the minimization process. This fact is used in the design example to constrain the stability robustness measure and actuator authority.

In the design example, we control the acoustic radiation from a submerged spherical shell. An analytical model is used to simulate measured (sampled) impulse responses. The plant has three disturbance sources, seven control sensors, eight control actuators, and 13 radiated field sensors. We require at least 10 dB H^∞ norm reduction in the radiated field, an H^∞ norm bound on the stability robustness measure, and an H^∞ norm bound on the closed-loop disturbance to individual-actuator transfer functions. This design requires that we simultaneously satisfy ten separate H^∞ constraints.

Several previous papers[3,7] suggest synthesizing multi-constraint H^∞ norm controllers by using an FIR plant model and an FIR Q-parameter: This paper provides a practical algorithm. Our algorithm is an adaptation of a basic technique from interior-point methods for convex programming.[12,14] Its main advantage, from a control designers point of view, is that multiple constraints simplify the translation from controlled-plant specifications to a controller synthesis problem. The design approach is more direct, and in principle, less experienced designers can develop controllers for larger and more complicated problems.

The body of this paper is organized into three parts: first, we discuss the plant and controller models, next we describe the controller design algorithm, and finally, we provide a brief design example.

---

† Some iterative methods that use approximate Newton directions retain polynomial complexity and rigorous stopping criteria.[13] Because of computer memory constraints, we are forced to use an iterative method that does not. This is explored in Section 3.3, Practical Considerations.

2. PLANT MODEL AND CONTROLLER PARAMETRIZATION

The plant and controller topology is shown in Figure 1. We assume that the plant is described by four finite-length impulse response matrices: **E** is the disturbance-to-regulated-output response, **S** is the disturbance-to-control-sensor response, **D** is the control-actuator-to-regulated-output-response, and **C** is the control-actuator-to-control-sensor response. The number of disturbance inputs is n_t, the number of control sensors is n_s, the number of control actuators is n_d, and the number of regulated outputs is n_r. Time series matrix elements will be denoted by superscript; that is, $E^{(j,k)}$ is the kth-disturbance-to-jth-regulated-output impulse response.

The controller is written in terms of two FIR filters: **N** and **Q**. We set the first to the opposite of the nominal control-actuator-to-control-sensor response, **N** = -**C**. This is the simplest possible Q-parametrization for stable plants. The control design parameters are the tap weights of **Q**, the finite-length impulse response representation of the Q-parameter. For most applications, we can assume that **C** has at least a sample delay, which facilitates the implementation of **N**.

Of course, the Q-parametrization has two attributes with make it a good choice for the design parameter. First, as Q ranges over all possible stable transfer functions, the controlled plant ranges over all possible stable closed-loop transfer functions. Second, many of the closed-loop quantities of interest are affine in **Q**; for example:

$$\mathbf{H}_1 = \mathbf{E} + \mathbf{D} \circledast \mathbf{Q} \circledast \mathbf{S}, \qquad \mathbf{H}_2 = \mathbf{Q} \circledast \mathbf{C} \qquad \text{and} \qquad \mathbf{H}_3 = \mathbf{Q}^{(1,:)} \circledast \mathbf{S} \tag{3}$$

where the first is the nominal closed-loop disturbance to regulated-output impulse response, the second is related to stability robustness and the third is the closed-loop disturbance to first-actuator response. The symbol ⊛ means discrete-time linear convolution. In the sequel, for notational simplicity, we write the closed-loop impulse responses associated with different constraints as

$$\mathbf{H}_i = \mathbf{E}_i + \mathbf{D}_i \circledast \mathbf{Q} \circledast \mathbf{S}_i \tag{4}$$

where $\mathbf{E}_i$, $\mathbf{S}_i$, $\mathbf{D}_i$, n_{ti}, and n_{ri} are specified for each constraint.

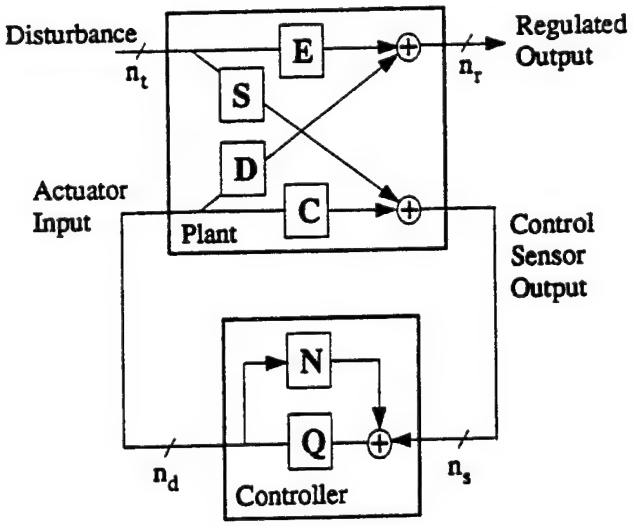

Figure 1. Plant and controller topology and notation

We apply H^∞ constraints to closed loop quantities that are affine in $\mathbf{Q}$. This ensures that the constraints, and their associated cost functions, are convex. It is this convexity that makes the optimization numerically tractable.

3. CONTROLLER DESIGN ALGORITHM

Standard H^∞ design methods try to find a controller that meets the equation (1) feasibility criterion. This paper's algorithm tries to find a controller that meets the feasibility criteria:

$$\max_k \bar{\sigma}(\mathbf{H}_i(e^{j\theta_k})) < \gamma_i \quad \text{for } k = 1 \text{ to } N_f \quad \text{and} \quad i = 1 \text{ to } N_c \tag{5}$$

where N_c is the number of separate transfer functions that are constrained. For systems of practical interest, for which the entire impulse response can be captured in $\mathbf{H}_i$, we expect $\max_k \bar{\sigma}(\mathbf{H}_i(e^{j\theta_k}))$ to approach $\sup_\theta \bar{\sigma}(\mathbf{H}_i(e^{j\theta}))$ non-monotonically as N_f grows.

If feasibility is determined, the standard (central) H^∞ controller design method minimizes the entropy integral:[15]

$$I_\gamma = \frac{\gamma^2}{2\pi}\int_{-\pi}^{\pi} -\log\det\left(\mathbf{I} - \frac{1}{\gamma^2}\mathbf{H}(e^{j\theta})^*\mathbf{H}(e^{j\theta})\right)\mathrm{d}\theta) \tag{6}$$

where γ is the entropy parameter, and $\mathbf{H}$ is some closed-loop transfer matrix. If the problem is infeasible, the integral is meaningless, and no controller is returned.

If feasibility is determined, our algorithm minimizes

$$\tilde{\phi}(\mathbf{Q}) = \sum_{i=1}^{N_c}\sum_{k=1}^{2(N_f-1)} -\log\det\begin{bmatrix}\gamma_i\mathbf{I} & \mathbf{H}_i(e^{j\theta_k}) \\ \mathbf{H}_i(e^{j\theta_k})^* & \gamma_i\mathbf{I}\end{bmatrix} \tag{7}$$

where the Schur compliment shows that the sum over k approximates (to a multiplicative and additive constant) the entropy integral. We refer to this sum as the generalized entropy, and we see that, for $N_c = 1$, a controller designed using these methods will approximate the central H^∞ controller. If the problem is infeasible, a controller that may approach feasibility is returned.

3.1. The cost functions

This algorithm uses two related cost functions. The first determines feasibility, and the second minimizes the generalized entropy once feasibility is established. The cost functions are minimized using Newton's method, which requires calculation of the gradient and Hessian.

The cost function used to determine feasibility is

$$\phi(t, \mathbf{Q}) = \sum_{i=1}^{N_c}\sum_{k=1}^{2(N_f-1)} -\log\det\begin{bmatrix}(\gamma_i + t\delta_i)\mathbf{I} & \mathbf{H}_i(e^{j\theta_k}) \\ \mathbf{H}_i(e^{j\theta_k})^* & (\gamma_i + t\delta_i)\mathbf{I}\end{bmatrix} + q\log t \tag{8}$$

where all variables are previously defined except for t, δ_i, and q. t is an auxiliary variable, and δ_i is a binary variable which applies t to only certain constraints. The cost function is initialized as follows:

1. Choose a starting value for $\mathbf{Q}$.

2. For each constraint, if

$$\begin{bmatrix} \gamma_i \mathbf{I} & \mathbf{H}_i(e^{j\theta_k}) \\ \mathbf{H}_i(e^{j\theta_k})^* & \gamma_i \mathbf{I} \end{bmatrix} > 0$$

for all k, then $\delta_i = 0$. If not, then $\delta_i = 1$.

3. Choose t so that the determinant's argument is positive definite for all k and i.

The Appendix proves that $\phi(t, \mathbf{Q})$ minimization has two possible outcomes: $t^* > 0$, which implies that the controller design problem is infeasible, and $\phi(t, \mathbf{Q})$ unbounded below, which determines a feasible $\mathbf{Q}$. The scale factor q is also described in the Appendix.

Once feasibility is established, we minimize the generalized entropy, $\tilde{\phi}(\mathbf{Q})$.

These cost functions have an alternative representation which is useful in calculating the Hessian and gradient and in proving the result in the Appendix. With the use of circulant matrices[16] and Kronecker products,[17] we can write

$$\phi(t, \mathbf{Q}) = -\log \det F(t, \mathbf{Q}) + q \log t \tag{9}$$

and

$$\tilde{\phi}(\mathbf{Q}) = -\log \det F(0, \mathbf{Q}) \tag{10}$$

where $F(t, \mathbf{Q}) = F_0 + F_t t + \sum_{ijk} F_k^{(ij)} \mathbf{Q}^{(ij)}(k)$, the F_0, F_t and $F_k^{(ij)}$ are symmetric matrices of dimension $2(N_f - 1) \sum_i (n_{ri} + n_{ii})$. The controller is feasible if and only if $F(0, \mathbf{Q})$ is positive definite, and, in equation (8), the determinant's argument is positive definite if and only if $F(t, \mathbf{Q})$ is positive definite. Minimizing $\tilde{\phi}(\mathbf{Q})$ is the well-known problem of finding the analytic centre of the linear matrix inequality, $F(0, \mathbf{Q}) > 0$, given a feasible starting point.[12]

In the following (and in the Appendix), we substitute $\mathbf{x}$ for $\mathbf{Q}$, where $\mathbf{x}$ has a single index, i, running from 1 to $n_s n_d l_q$. From Boyd and El Ghaoui,[12] or Nesterov and Nemirovskii,[14] the Hessian and gradient of $-\log \det F(t, \mathbf{x})$ are written:

$$\begin{aligned} \mathcal{G}_t &= -\text{trace}\{V F_t\} \\ \mathcal{G}_i &= -\text{trace}\{V F_i\} \\ \mathcal{H}_{tt} &= \text{trace}\{V F_t V F_t\} \\ \mathcal{H}_{ti} &= \text{trace}\{V F_t V F_i\} \\ \mathcal{H}_{i'i} &= \text{trace}\{V F_{i'} V F_i\} \end{aligned}$$

where $V = F(t, \mathbf{x})^{-1}$, and $\mathcal{H}_{it} = \mathcal{H}_{ti}$. After including the $q \log t$ term in the gradient only, the Newton direction is, of course, $-\mathcal{H}^{-1}\mathcal{G}$. The gradient and Hessian for $\tilde{\phi}$ are computed with the same method, using only $\mathcal{G}_i$ and $\mathcal{H}_{i'i}$.

3.2. *Design algorithm*

Step 0. Assemble the $\mathbf{E}_i$, $\mathbf{S}_i$, $\mathbf{D}_i$ and γ_i. *Choose* l_q (the length of the Q-parameter FIR) and q. Initialize $\phi(t, \mathbf{Q})$.

Step 1. Use Newton's method on ϕ to determine a search direction in $(t, \mathbf{Q})$, and use a guarded line search to determine the minimizer in that direction. The guards are set so that the updated value of $F(t, \mathbf{Q})$ remains positive definite, but they ignore the constraint on t implied by ϕ's second term. If the updated value of t is negative at any point during this line search,

$q(t, \mathbf{Q})$ is unbounded below, the associated $\mathbf{Q}$ satisfies equation (5), and we can proceed immediately to Step 2. If t remains positive, we update $(t, \mathbf{Q})$ and repeat this step. Optionally, we can update δ_i at any point in the line search. If a constraint becomes feasible, we can set the appropriate $\delta_i = 0$, which captures that constraint and ensures its continued feasibility. If Newton's method converges (which implies that ϕ's minimizer has a positive t), then at least one of the chosen γ_i is infeasible, the algorithm ends, and the returned controller meets the criteria

$$\max_k \bar{\sigma}(\mathbf{H}_i(e^{j\theta_k})) < (\gamma_i + t\delta_i) \quad \text{for } k = 1 \text{ to } N_f \quad \text{and} \quad i = 1 \text{ to } N_c \tag{11}$$

The controller is optimal in the sense that it minimizes ϕ.

Step 2. Starting at the feasible $\mathbf{Q}$ from Step 1, we use Newton's method with guarded line search on $\bar{\phi}$, which converges to the minimum of equation (7).

Efficient methods for the guarded line search are available, as well as a rigorous stopping criterion for Step 2.[12,14]

In Step 0, we require the designer to choose the length of the Q parameter. It is not clear *a priori* how long the FIR has to be. Experience indicates that a good first choice is the length of the plant data. For any choice of l_q, if Step 1 determines a feasible controller, the design process can be halted. If Step 1 does not determine a feasible controller, and if the Q-parameter time series do not decay to zero, we can gain design freedom by lengthening $\mathbf{Q}$.

3.3. Practical considerations

Applying this method brings up several questions of practical concern. 'Will data scaling aid convergence?' 'What is the best method for finding the Newton direction?' And 'what are the effects of using an approximate Newton direction?'

3.3.1. Data scaling. The constraint applied to transfer matrix $\mathbf{H}$ does not change if equation (2) is multiplied by any positive constant. The cost functions, however, will vary in shape and minimizer. This scaling can accelerate convergence by concentrating the Newton steps in the directions most in need of minimization. It can also decelerate convergence by forcing the minimizer up against one (or more) of the constraints and reducing the possible distance travelled in the line search. An example of this latter property is shown in the design example. We have not performed a systematic study of balancing the two considerations.

Additionally, q can be chosen to be any positive constant greater than the dimension of F. We choose $q = 5 \dim(F)$, but we have not performed a systematic study of this scale factor's effect.

3.3.2. Calculating the Newton direction. Most problems of practical size are fairly large, with 10^3 to 10^5 free parameters in the Q discretization. This makes direct assembly of the Hessian and gradient and direct calculation of the Newton direction very difficult.

The structure in F gives the Hessian structure that can be exploited. Using DFTs[16] to block diagonalize the F matrix and Kronecker product identities to further order the calculation,[17] we can form the matrix–vector product, $\mathcal{H}\mathbf{v}$, in $\mathcal{O}(N_c N_f l^3 + N_f \log_2(N_f) l^2)$ operations without forming the Hessian. (For simplicity, we assume that $n_{ti} = n_{si} = n_d = n_r = l$.) This economical matrix–vector multiplication suggests using the conjugate gradient (CG) algorithm to solve for the Newton direction.[18] Even with this economical matrix–vector multiply, the CG algorithm is

not well-suited to finding the exact Newton direction: the Hessian is often ill-conditioned, and finite precision arithmetic causes loss of orthogonality. The former slows convergence, and the latter can stop it. To alleviate the finite-precision problem, we orthogonalize the search directions and residuals and iterate until a fixed convergence criterion or a fixed maximum number of iterations is reached. Limited memory requires the fixed maximum to be set small compared with the dimension of $\mathcal{H}$. Other methods for approximating the Newton direction—including preconditioned CG, incomplete orthogonalization, and regularization of $\mathcal{H}^{-1}$—are certainly possible but have not been systematically investigated.

3.3.3. Effects of an approximate Newton direction. The rigorous stopping criterion for Step 2 mentioned in the previous section holds only for exact Newton directions. When using an approximate Newton direction, Δx, experience indicates that we can stop the Newton iterations when $\mathcal{G}^T \Delta x$ is sufficiently small. Of course, 'sufficiently small' depends on the problem data. We use the same stopping criterion in Step 1, although the concave term in ϕ makes the criterion heuristic even for exact Newton directions.

The heuristic stopping criteria imply that negative results are not definitive. If Step 1 appears to converge to a positive t or stalls without convergence, this does not imply that there are no feasible controllers, and restarting the algorithm with a different data scaling may prove worthwhile. Regardless, a $\mathbf{Q}$ is returned that attains H^∞ bounds of $\gamma_i + \delta_i t$ instead of the desired γ_i. Another useful way to modify a negative result is to initialize a second application of the algorithm with the returned $(t, \mathbf{Q})$, and replace γ_i with $\gamma_i + \delta_i(t + \varepsilon)$. In this second application, the initial $\mathbf{Q}$ is feasible, and we move directly to Step 2, minimizing the generalized entropy. For small ε and appropriate weighting, the minimization will 'push' down the H^∞ norms of the infeasible constraints. This method is explored in the design example.

Limited computer memory requires that we calculate the approximate Newton direction using a fixed number of CG iterations. The CG method proposed by Vandenberghe and Boyd[13] halts the iterations based on a convergence criterion alone. They show that attaining their criterion guarantees a constant reduction in cost, which implies polynomial complexity. Using a fixed number of iterations destroys this result.

Regardless of the heuristic stopping criteria and the loss of polynomial complexity, experience shows that this method can be used to design controllers that satisfy multiple H^∞ norm constraints for realistically sized systems. Of particular utility is the fact that if a constraint is met for the initial $\mathbf{Q}$, it will not be exceeded in the subsequent optimization. This is shown in following example.

4. MULTI-SENSOR, MULTI-ACTUATOR CONTROL OF SOUND RADIATED FROM A SUBMERGED SPHERICAL SHELL WITH STABILITY ROBUSTNESS AND INDIVIDUAL ACTUATOR AUTHORITY CONSTRAINTS

This problem serves two purposes: It shows that the algorithm presented above can be practically applied to realistically sized problems, and it provides two design examples for the new method. The first example shows a successful design, and the second demonstrates failure due to stalled convergence.

With reference to Figure 2, the disturbance is modelled as three internal normal-force sources located asymmetrically around the sphere's north pole $(+z)$. Seven control sensors are placed on the shell in the region adjacent to the disturbance, seven control actuators are arrayed on the shell in the region adjacent to the sensors, and one control actuator is placed on the sphere's south pole. These regions are circularly symmetric, but the elements are asymmetrically placed.

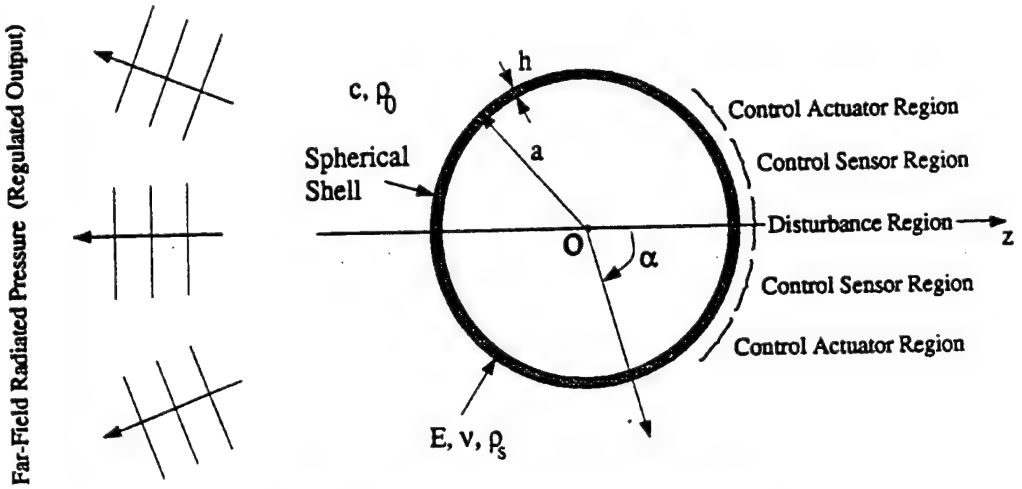

Figure 2. Physical configuration for the design example

Finally, 13 far-field pressure sensors are arrayed in a 60° sector around the sphere's south pole. Table 1 gives the exact element locations, where α is degrees latitude, measured from the $+z$ axis, and β is degrees longitude, measured from the $+x$ axis (not shown in Figure 2).

We wish to control the radiation from the three disturbance sources to the 13 far-field pressure sensors. The first H^{∞} requirement states that the minimum reduction in maximum disturbance-to-regulated-output singular value is 10 dB, that is,

$$\max_k \bar{\sigma}\{\mathbf{E}(e^{j\theta_k}) + \mathbf{D}(e^{j\theta_k})\mathbf{Q}(e^{j\theta_k})\mathbf{S}(e^{j\theta_k})\} \leq 0{\cdot}316 \max_k \bar{\sigma}\{\mathbf{E}(e^{j\theta_k})\} \tag{12}$$

With reference to equations (4) and (5), we take $\mathbf{E}_1 = \mathbf{E}$, $\mathbf{D}_1 = \mathbf{D}$, $\mathbf{S}_1 = \mathbf{S}$, and $\gamma_1 = 0{\cdot}316 \max_k \bar{\sigma}[\mathbf{E}(e^{j\theta_k})]$. Of course, weighting matrices can be applied to emphasize certain frequency regions and combinations of disturbance and regulated output.

We assume a multiplicative matrix perturbation at the actuator input, and require that the system be guaranteed stable for perturbations with maximum singular value less than or equal to

Table I. Element locations (α, β) in degrees

	disturbance	Control sensor	Control actuator	FF pressure
1	(0,0)	(17,175)	(35,0)	(180,0)
2	(5,0)	(15,280)	(30,0)	(175,0)
3	(10,135)	(10,315)	(180,0)	(175,90)
4		(25,0)	(30,240)	(175,180)
5		(20,35)	(35,135)	(175,270)
6		(25,90)	(30,135)	(150,0)
7		(20,225)	(37,41)	(150,45)
8			(33,170)	(150,90)
9				(150,135)
10				(150,180)
11				(150,225)
12				(150,270)
13				(150,315)

one. The second H^{∞} constraint requires that

$$\bar{\sigma}(\mathbf{Q}(e^{i\theta_k})\mathbf{C}(e^{i\theta_k})) \leq 1 \quad \text{for } k = 1 \text{ to } N_f \tag{13}$$

We assume no frequency shaping on this requirement. With reference to equations (4) and (5), we take $\mathbf{E}_2 = 0$, $\mathbf{D}_2 = \mathbf{I}$, $\mathbf{S}_2 = \mathbf{C}$ and $\gamma_2 = 1$, where $\mathbf{I}$ is the identity matrix impulse response.

We also assume that each actuator is power limited, and that the 2-norm of the disturbance-to-individual-actuator transfer vector is a good estimate of (the root of) that power. The third to tenth H^{∞} norm constraints require that

$$\bar{\sigma}(\mathbf{Q}^{(j,:)}(e^{i\theta_k})\mathbf{S}(e^{i\theta_k})) \leq 1 \quad \text{for} \quad k = 1 \text{ to } N_f, \quad j = 1 \text{ to } n_d \tag{14}$$

which limits the estimate of the power to a numerical value of one. With reference to equations (4) and (5), we take $\mathbf{E}_i = 0$, $\mathbf{D}_i = \mathbf{e}_{(i-2)}$, $\mathbf{S}_i = \mathbf{S}$ and $\gamma_i = 1$, where $\mathbf{e}_j$ is the jth row of $\mathbf{I}$. Again, weighting matrices can be used to shape the frequency response.

The remainder of this section includes a brief description of the submerged shell model and a discussion of the design procedure and results.

4.1. Spherical shell model and control configuration

Again, with reference to Figure 2, we define the speed of sound in the exterior fluid, $c = 1\,480$ m/s; the fluid's density, $\rho_0 = 1\,000$ kg/m^3; the sphere's radius, $a = 5$ m; thickness, $h = 0{\cdot}15$ m; Young's modulus, $E = 19{\cdot}6 \times 10^{10} \times (1 + 0{\cdot}06i)$ N/m^2; Poisson's ratio, $\nu = 0{\cdot}3$; and

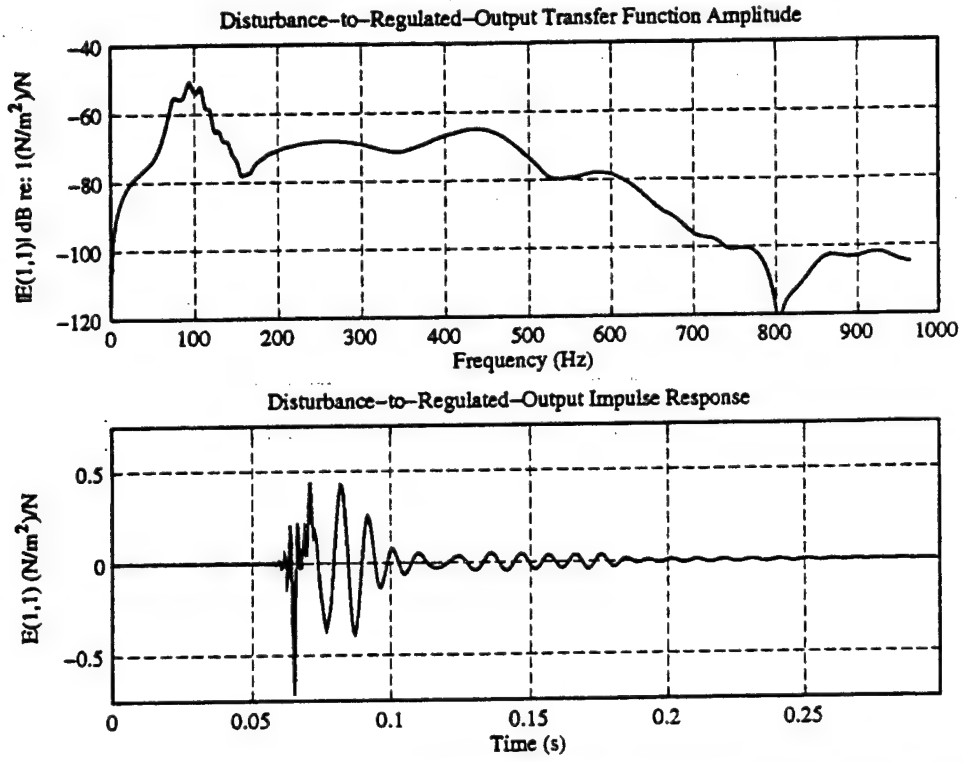

Figure 3. Typical (open-loop) disturbance to regulated-output transfer function and impulse response. The delay corresponds to the 100 m transmission path. Note the flat amplitude response and associated high-frequency early-time impulse response

material density, $\rho_s = 7\,668{\cdot}7$ kg/m^3. Complex Young's modulus is included to simulate material damping in the shell. The regulated outputs are the far-field radiated pressure referenced to 100 m. We assume that the control sensors measure normal (outward) velocity and that the actuators provide a normal force. The dynamics of the disturbance sources and the actuators are assumed to be those of an eighth-order low-pass Butterworth filter with cutoff at 500 Hz. The actuator and disturbance dynamics also serve as anti-alias filters.

We use the standard bending shell approximation, as presented in Junger and Feit,[19] for the spherical shell, coupled to an exact solution for the infinite-extent fluid. The normal-force-to-normal-velocity transfer function is given by Junger and Feit, equation (9.13), and the normal-force-to-far-field-pressure, referenced to 1 m, is given by equation (9.14). We calculate the transfer functions at 2 049 evenly-sampled frequencies from 0 to 965·28 Hz and then window with the actuator/disturbance dynamics. At 965 Hz, which we assume to be the Nyquist frequency, the amplitudes are ~40 dB down from their maxima. At zero frequency (a degenerate point in the mathematical model), as at the imposed Nyquist, the transfer functions are taken to be 0.

The impulse responses are calculated by 4 096-point (unscaled) FFT. Using the actuator dynamics to roll off the transfer functions minimizes acausal leakage due to the frequency windowing.

The (open-loop) $\mathbf{E}^{(1,1)}$ disturbance-to-regulated-output transfer function amplitude and impulse response are shown in Figure 3. The (open-loop) $\mathbf{S}^{(1,1)}$ disturbance-to-control-sensor transfer function amplitude and impulse response for are shown in Figure 4. These traces are

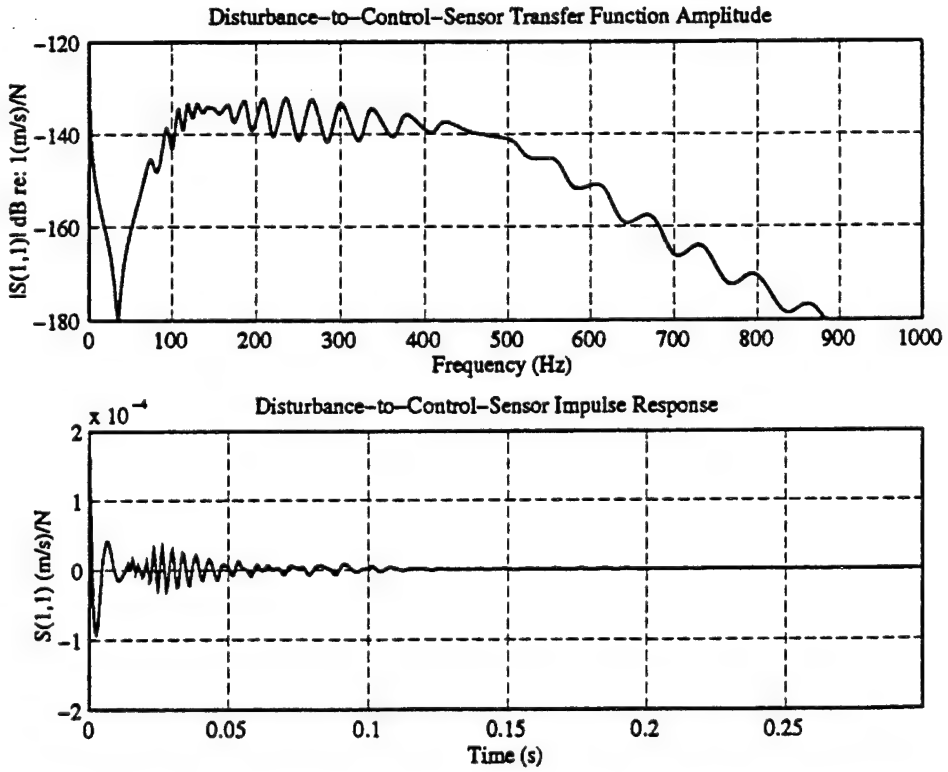

Figure 4. Typical (open-loop) disturbance to control-sensor transfer function and impulse response

typical of **E** and **D** data and of **S** and **C** data, respectively: the transfer functions have no obvious modal decomposition, and long-time ringing is absent from the impulse responses.

4.2. Controller design

We give two design procedure examples. The first starts from a zero **Q**, which satisfies the stability robustness and actuator authority constraints, but violates the radiation reduction criterion. The stability robustness and actuator authority constraints remain satisfied throughout the design, and we try to determine feasibility for the radiation reduction. This design is successful: 10 dB radiation reduction is achieved, and the generalized entropy is minimized. In the second design, we initialize our algorithm with the **Q** resulting from an H^2 method. This **Q** satisfies the radiation reduction criterion but violates the other nine. The radiation reduction criterion stays satisfied throughout the design, and we try to determine feasibility for the others. This design is unsuccessful: the Newton iterations stall in Step 1. This result is partially rectified by the method suggested in Section 3.3: γ_i is replaced by $\gamma_i + \delta_i(t+\varepsilon)$ and the algorithm is restarted at Step 2. All but one of the constraints are feasible after 30 additional Newton iterations.

In all controller designs, we use the first 800 taps of the plant data in forming the cost functions. Figures 3 and 4 show that the **E**, **S**, **D** and **C** are easily captured in 800 tap (~0·4 s) impulse responses. We assume that 400 taps will suffice to represent the entire Q-parameter, and we take $2(N_f - 1) = 2\,048$, which is the smallest power of two that contains the design closed-loop impulse response. The H^∞ design steps require that we minimize ϕ and $\bar{\phi}$ over $7 \times 8 \times 400 + 1 = 22\,401$ and 22 400 variables, respectively.

For simulation, we use 1 024-point plant impulse responses, and compute frequency domain results with 4 096-point FFTs. After 1 024 taps, the amplitudes of **E** and **D** are down 100 dB from their maxima and the amplitudes of **S** and **C** are down 50 dB. We assume that successful simulation represents successful control of the modelled plant.

4.2.1. H^∞ design, starting at $Q = 0$. In this H^∞ design, we initialize with $\mathbf{Q} = 0$. The stability robustness and actuator authority constraints are initially satisfied. Radiation reduction is not satisfied, so we choose the initial t so that $\gamma_1 + t$ is greater than the H^∞ norm of $\mathbf{E}_1$. We take $\gamma_1 = 16 \approx 0{\cdot}316 \max_k \bar{\sigma}\{\mathbf{E}(e^{j\theta_k})\}$, and in order to emphasize the radiation reduction requirement, we scale the first constraint by 100. The starting $t \approx 4\,873$. The Newton iteration stopping criterion is 10^{-5}, and the maximum number of iterations allowed is 30. Similarly, 30 iterations of the guarded line search are also allowed.

With over 2×10^4 free variables, exact solution of the Newton direction is impractical, and truncated CG with full orthogonalization is used to approximate $\mathcal{H}^{-1}\mathcal{G}$. The CG iterations are terminated when the standard tolerance parameter decreases to 10^{-5} or after a maximum of 30 iterations.

The design was successful. The results plotted in Figure 5 focus on the frequency region with the most activity, 0–200 Hz, out of a possible 0–965 Hz. The top plot shows the maximum singular values for the controlled and uncontrolled disturbance-to-regulated-output. The 10 dB norm-reduction goal is met. The middle plot shows the maximum singular values of the stability robustness measure, and the bottom shows the maximum singular values of disturbance-to-individual-actuator transfer functions 1, 2 and 3. The norm goals for these transfer functions are also met.

Figure 6 shows the H^∞ norms' evolution as the algorithm progresses. The algorithm exits after 30 Newton iterations with a stopping criterion of $\sim 8{\cdot}3 \times 10^{-2}$. The evolution stops after 20

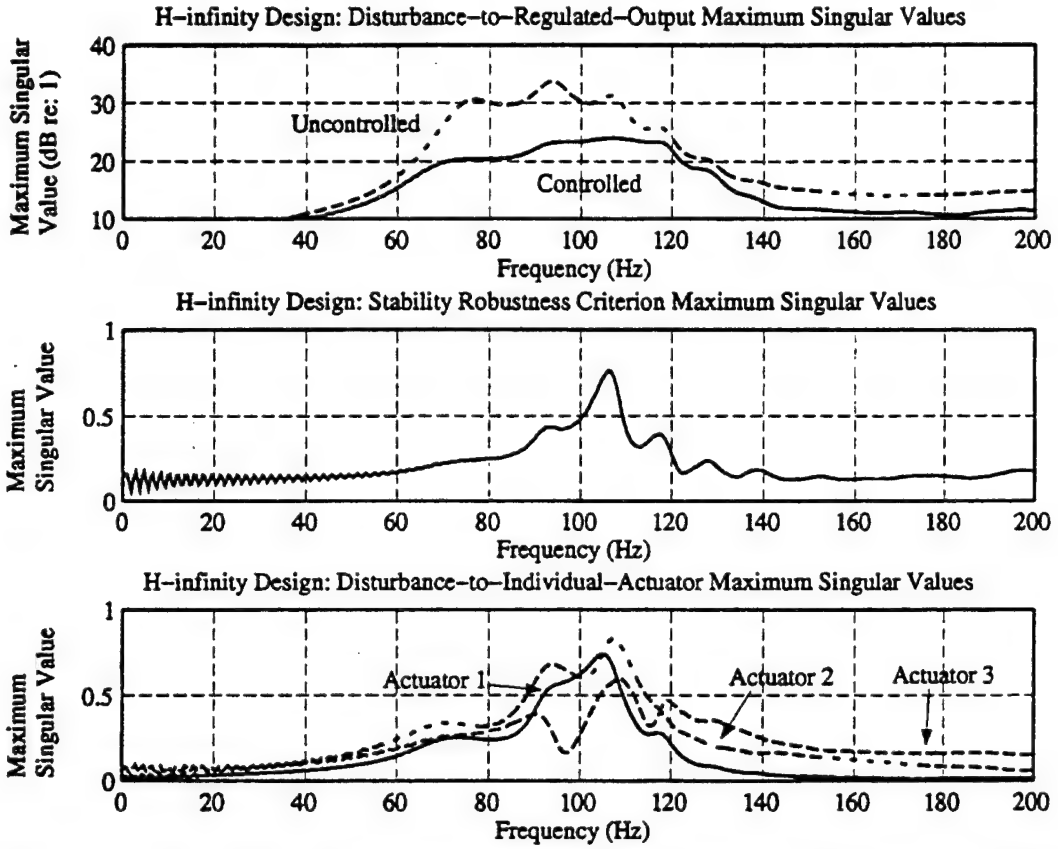

Figure 5. H^{∞} design maximum singular value plots disturbance-to-regulated-output (controlled and uncontrolled), stability robustness measure and disturbance-to-individual-actuators 1, 2 and 3.

(plotted) iterations. The top plot shows the maximum singular value of the disturbance to regulated output. The dashed horizontal line indicates the desired γ_1. Step 1 determined a feasible controller after 11 iterations. After that, Step 2 looked for the minimum generalized entropy controller. The middle and bottom plots show the H^{∞} norm evolution of the stability robustness and the actuator authorities, respectively.

4.2.2. H^{∞} design, starting at the H^2 Q. The H^2 design is accomplished using code developed by Olkin *et al.*[11] which yields the $\mathbf{Q}^*$ that minimizes

$$\sum_{i=1}^{n_r}\sum_{j=1}^{n_i}\sum_{k}\{\mathbf{H}^{(i,j)}(k)\}^2 + 0{\cdot}01\mu\sum_{i=1}^{n_i}\sum_{j=1}^{n_z}\sum_{l=0}^{l_q-1}\{\mathbf{Q}^{(i,j)}(l)\}^2 \tag{15}$$

where k runs from zero to the closed-loop impulse response length, and

$$\mu = \max_{i'j'l'}\frac{\partial^2}{\partial(\mathbf{Q}^{(i',j')}(l'))^2}\sum_{i=1}^{n_r}\sum_{j=1}^{n_i}\sum_{k}\{\mathbf{H}^{(i,j)}(k)\}^2$$

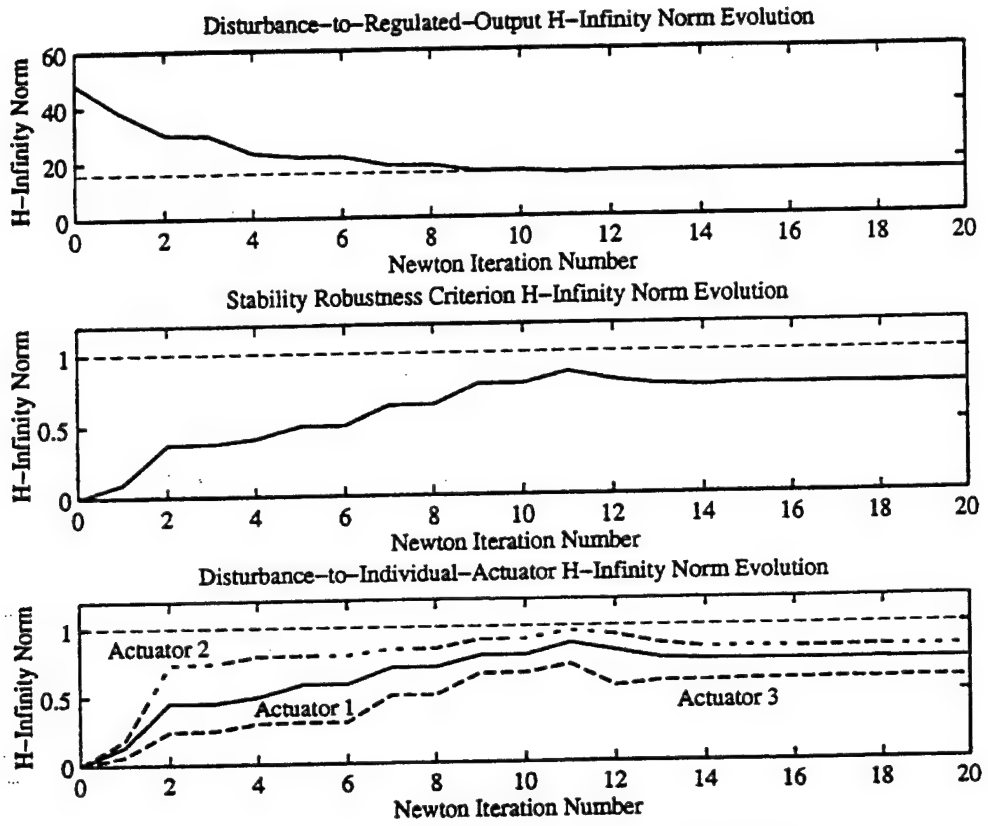

Figure 6. Evolution of the maximum singular values for the successful H^{∞} design

We take **H** to be the closed-loop disturbance-to-regulated-output impulse response, and the second term in (15) regularizes the solution by weighting the 2-norm of **Q** by 0·01 μ. The form of μ scales the weighting to the problem at hand, and the assignment of the scale factor (0·01) is entirely heuristic: experience indicates that this level of regularization gives good performance at the sacrifice of some robustness.

The results of this design are shown in Figure 7. The top plot shows the maximum singular values for the uncontrolled and controlled disturbance-to-regulated-output transfer function. This **Q** gives us about a 25 dB reduction in H^{∞} norm, indicating that the 10 dB reduction required of the system should be well within reach. The middle plot shows the maximum singular values for the stability robustness measure, and the bottom shows the maximum singular values for disturbance-to-individual-actuator transfer functions 1, 2 and 3. These traces are well above the desired bound (of 1) and indicate that the H^{∞} algorithm will have work to do.

The second H^{∞} design starts from the H^2 result. The radiation reduction requirement is initially satisfied and will remain so throughout. The stability robustness and actuator authority constraints are not initially satisfied, so we must choose the initial t so that the $\gamma_i + t$ is feasible for constraints 2–10. To emphasize the constraints that are not satisfied, we scale their data by a factor of 100. The starting $t \approx 3\ 723$.

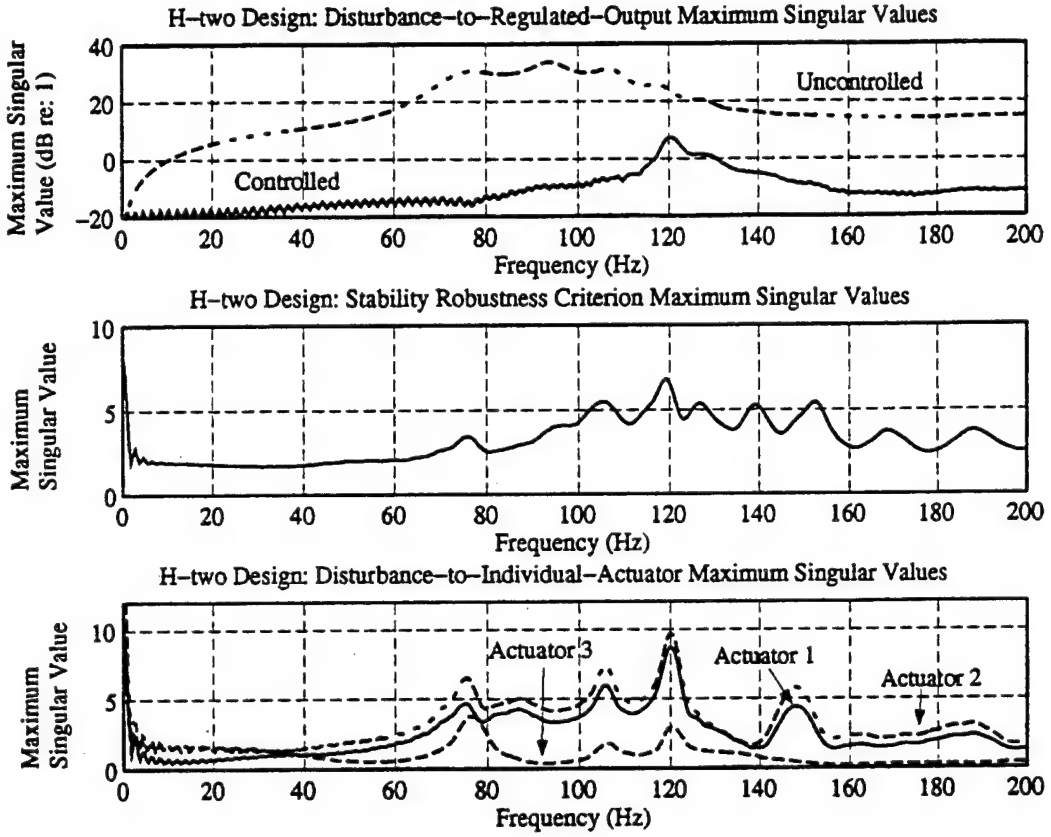

Figure 7. H^2 design maximum singular value plots disturbance-to-regulated-output (controlled and uncontrolled), stability robustness measure and disturbance-to-individual-actuators 1, 2 and 3

This design, which we know has a feasible controller, was unsuccessful. The norms' progress with Newton iteration is shown in Figure 8. On exit, the stopping criterion was oscillating and had not begun to converge. We see that the disturbance-to-regulated-output remains bounded, and that the others cease to progress after the disturbance-to-regulated-output is pinned against its bound. We note that the three plotted disturbance-to-individual-actuator singular values *do* become feasible at iteration 11: the algorithm modifies δ_i, and they are captured in the feasible region. The stability robustness and disturbance-to-individual-actuator transfer functions four and six stay infeasible. This design fails because poor data scaling makes the convergence rate unacceptably slow.

The effort expended in computing this controller is not wasted: we have achieved the H^{∞} bounds indicated by the final data points of Figure 8. These results also indicate how to proceed in subsequent minimization attempts. For example, we replace γ_i with $\gamma_i + (t + 3 \times 10^{-3})\delta_i$ to make all the constraints (barely) feasible and initialize with the failed H^{∞} $(t, \mathbf{Q})$. After 30 Newton iterations, all but actuator authority constraint four are satisfied. Of course, this method can be applied again, focusing on the remaining infeasible constraint.

These multiple H^{∞} norm designs demonstrate this algorithm's utility, even with the approximate Newton directions. The special purpose H^{∞} design software is written in

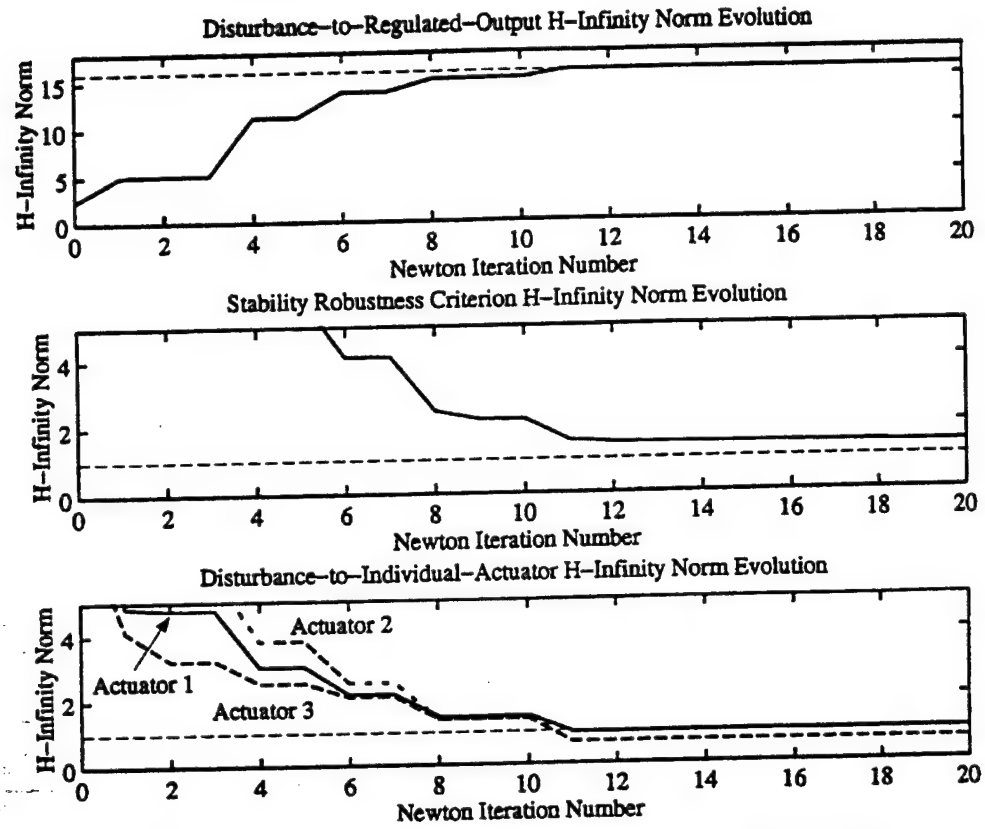

Figure 8. Evolution of the maximum singular values for the unsuccessful H^{∞} design

MATLAB, and each of these designs took between 10 and 15 hours on a Sun SPARC 10 workstation. We expect FORTRAN or C translation to significantly speed the solutions.

5. CONCLUSIONS

The practical algorithm presented in this paper allows us to specify and simultaneously satisfy multiple H^{∞} norm constraints. This capability greatly increases the utility of H^{∞} norm concepts in controller synthesis, as demonstrated in the design example. We approximate the standard H^{∞} norm by the maximum singular value of the sampled transfer matrices, and we can constrain any closed-loop transfer function that is affine in the Q parameter. FIR matrices describe the plant and the Q-parameter. This is a natural description for many finite- and infinite-dimensional systems and can be taken directly from measured data. The tap weights of Q are the controller design parameters, and hardware implementation of $\mathbf{Q}$ and $\mathbf{N}$ is straightforward.

The algorithm proceeds in two steps. First, we determine multiple constraint feasibility; if feasibility is determined, we then minimize the generalized entropy. The first step is a result of the 'theorem of alternatives' proved in the Appendix, and the second step finds the analytic center of the given linear matrix inequality. The second step is an element of the method of centres, an interior-point method for solving more general convex problems. Issues of practical

concern in implementation are also discussed, the most important of which is the effect of using approximate Newton directions.

Two 10-constraint design examples are also provided. We control the acoustic radiation from a submerged spherical shell while maintaining stability robustness and eight individual actuator authority constraints. The first design is immediately successful. The second requires iterative minimization. The required iteration demonstrates one of the effects of approximate Newton directions.

The major difficulty with this algorithm is the lack of a rigorous stopping criterion. We have no way of knowing whether we have reached the optimum, or whether the minimization procedure has merely stalled. This suggests finding an optimization method that uses a cost function with a known minimum. The primal–dual potential reduction method for semi-definite programming[20] is such a method. We are currently investigating this method for use in an improved multi-constraint H^∞ norm controller synthesis code.

APPENDIX. THEOREM OF ALTERNATIVES FOR $\phi(t, \mathbf{Q})$ MINIMIZATION

An outline of this proof was provided by Stephen Boyd.

Background. Given symmetric $F(t,x) = F_0 + F_t t + \sum F_i x_i > 0$, and $t > 0$, we define the cost function

$$\phi(t, \mathbf{x}) = -\log \det F(t, \mathbf{x}) + q \log t \tag{16}$$

The first term is convex, rising to infinity along $(0, \mathbf{x})$ lines as F approaches positive-semi-definite. The second term is concave, falling to $-\infty$ as t approaches 0. We choose $q > \dim(F)$ to guarantee that $\phi \to \infty$ as $t \to \infty$. The ϕ minimization has two alternatives: first, if the minimizer of $\phi(t, \mathbf{x})$ occurs at $(t^*, \mathbf{X}^*)$, then we can use the optimality conditions to show that, if $t^* > 0$, then $F(0, \mathbf{x})$ is non-positive-definite for all $\mathbf{x}$; second, with $q > \dim(F)$, if ϕ is unbounded below, then $F(0, \mathbf{x}) > 0$ for some $\mathbf{x}$.

Proof. We want to show: if $t^* > 0$, then $F(0, \mathbf{x})$ is non-positive-definite for all $\mathbf{x}$. The second alternative is evident from equation (16).

Assume that $t^* > 0$, then we can use the optimality conditions,

$$\begin{aligned} \frac{\partial \phi}{\partial t}(t^*, \mathbf{x}^*) &= -\text{trace}\,\{F(t^*, \mathbf{x}^*)^{-1} F_t + \frac{q}{t^*} = 0 \\ \frac{\partial \phi}{\partial x_i}(t^*, \mathbf{x}^*) &= -\text{trace}\,\{F(t^*, \mathbf{x}^*)^{-1} F_i\} \qquad = 0 \end{aligned} \tag{17}$$

to write the two equivalent expressions

$$\text{trace}\{F(t^*, \mathbf{x}^*)^{-1} F(t^*, \mathbf{x})\} = \text{trace}\{F(t^*, \mathbf{x}^*)^{-1} F(t^*, \mathbf{x}^*)\} \tag{18}$$

$$= \dim(F) \tag{19}$$

$$\text{trace}\{F(t^*, \mathbf{x}^*)^{-1} F(t^*, \mathbf{x})\} = \text{trace}\{F(t^*, \mathbf{x}^*)^{-1} F_t\} t^* + \text{trace}\{F(t^*, \mathbf{x}^*)^{-1} F(0, x)\}\} \tag{20}$$

$$= q + \text{trace}\{F(t^*, x^*)^{-1} F(0, \mathbf{x})\} \tag{21}$$

Subtracting, we find that

$$\text{trace}\{F(t^*, \mathbf{x}^*)^{-1} F(0, \mathbf{x})\} = \dim(F) - q \tag{22}$$

and so our choice $q > \dim(F)$ yields $\text{trace}\{F(t^*, \mathbf{x}^*)^{-1} F(0, \mathbf{x})\} < 0$. Because $F(t^*, \mathbf{x}^*) > 0$, the trace inequality implies that $F(0, \mathbf{x})$ is non-positive-definite for all x. □

ACKNOWLEDGEMENTS

I am grateful to Stephen Boyd, David Flamm and Julia Olkin for instructive conversations throughout the preparation of this material. I am also grateful to Anton Stoorvogel and the anonymous reviewers for their many comments and corrections. This work is supported in part by SRI International.

REFERENCES

1. Boyd, S. P., V. Balakrishnan, C. H. Barratt, N. M. Khraishi, Xi. Li, D. G. Meyer and S. A. Norman, 'A new CAD method and associated architectures for linear controllers,' *IEEE Transactions on Automatic Control*, **33**(3), 268–283 (1988).
2. Boyd, S. and C. Barratt, *Linear Controller Design: Limits of Performance*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
3. Scherer, C., 'Multiobjective H^2/H^∞ control,' *IEEE Transactions on Automatic Control*, *40*, 1054–1062 (1995).
4. Doyle, J. C., K. Glover, P. P. Khargonekar and B. A. Francis, 'State-space solutions to standard H^2 and H^∞ control problems', *IEEE Transactions on Automatic Control*, **34**(8), 831–847 (1989).
5. Lublin, L. and M. Athans, 'Experimental comparison of h^2 and h^∞ designs for a flexible structure', *Proceedings of the 33rd Conference on Decision and Control*, 1994, pp. 2910–2915.
6. Vidyasagar, M., *Control System Synthesis: A Factorization Approach*, The MIT Press, Cambridge, Massachusetts,1985.
7. Boyd, S., C. Barrat and S. Norman, 'Linear controller design: limits of performance via convex optimization', *Proceedings of the IEEE*, **78**(3), 529–573 (1990).
8. Peterson, D. K., W. A. Weeks and W. C. Nowlin, 'Active control of complex noise problems using a broadband, multichannel controller', *Noise-Con 94*, New York, 1994, pp. 315–320.
9. Weeks, W. A. and B. L. Curless,. 'A real-time multichannel system with parallel digital signal processors,' *Proceedings of the 1990 International Conference on Acoustics, Speech, and Signal Processing*, New York, 1990, pp. 1787–1790.
10. Weeks, W. A., C. T. Chittenden and P. M. Heilman, 'Hardware control of a real-time multichannel digital signal processing system', *Proceedings of the 25th Asilomar Conference on Signals, Systems, and Computers*, Los Alamitos, California, 1991, pp. 423–428.
11. Olkin, J. A., M. Freed, C. Gellrich and P. Jungers, *SWAPS User's Manual*, System Technology Division, SRI International, 1994.
12. Boyd, S. and L. El Ghaoui, 'Method of centers for minimizing generalized eigenvalues,' *Linear Algebra and Applications*, July 1993.
13. Vandenberghe, L. and S. Boyd, 'Primal–dual potential reduction method for problems involving matrix inequalities,' To appear in *Mathematical Programming Series B*, 1995.
14. Nesterov, Y. and A. Nemirovsky, *Interior Point Polynomial Methods in Convex Programming: Theory and Application*, Springer-Verlag, 1993.
15. Mustafa, D. and K. Glover, *Minimum Entropy H^∞ Control*, Springer-Verlag, New York, 1990.
16. Davis, P. J., *Circulant Matrices*, John Wiley and Sons, New York, 1979.
17. Graham, A., *Kronecker Products and Matrix Calculus: with Applications*, John Wiley and Sons, New York, 1981.
18. Golub, G. H. and C. F. van Loan, Matrix Computations, The John Hopkins University Press, Baltimore, 1989, pp. 49–69.
19. Junger, M. C. and D. Feit, *Sound, Structures, and Their Interaction*, The MIT Press, Cambridge, MA, 1986.
20. Vandenberghe, L. and S. Boyd, 'Semi-definite programming', To appear in *SIAM Review* (1995).

Semi-definite programming for quadratically constrained quadratic programs

Julia Olkin
Paul Titterton, Jr.

Applied Control and Signal Processing Group, SRI International
Menlo Park, CA 94025

1 ABSTRACT

We consider the linear least squares problem subject to multiple quadratic constraints, which is motivated by a practical application in controller design. We use the techniques of convex optimization, in particular, interior-point methods for semi-definite programming.

We reduce a quasi-convex potential function. Each iteration requires calculating a primal and dual search direction and minimizing along the plane defined by these search directions. The primal search direction requires solving a least squares problem whose matrix is composed of a block-Toeplitz portion plus other structured matrices. We make use of Kronecker products and FFT's to greatly reduce the calculation. In addition, the matrix updates and matrix inverses in the plane search are actually low-rank updates to structured matrices so we are able to further reduce the flops required. Consequently, we can design controllers for problems of considerable size.

2 INTRODUCTION

The problem this paper addresses is

$$\begin{array}{ll} \min\limits_{\mathbf{W}} & \|\mathbf{E}_0 + \mathbf{D}_0 \circledast \mathbf{W} \circledast \mathbf{S}_0\|_2 \\ \text{subject to} & \|\mathbf{E}_j + \mathbf{D}_j \circledast \mathbf{W} \circledast \mathbf{S}_j\|_2 \leq \alpha_j, \ j = 1, \ldots, k. \end{array} \tag{1}$$

We refer to (1) as Problem QCQP, for quadratically constrained quadratic programming. The data $(\mathbf{E}, \mathbf{S}, \mathbf{D})$ are finite duration matrix functions of discrete time where $\mathbf{E}$ is (n_r, n_t, l_e), $\mathbf{D}$ is (n_r, n_d, l_d) and $\mathbf{S}$ is (n_s, n_t, l_s), the design variable $\mathbf{W}$ is (n_d, n_s, l_w), and $\circledast$ is the discrete time convolution operator. The 2-norm of an impulse response matrix is

$$\|\mathbf{H}\|_2 = \sum_n \sum_{i,j} (h_{ij}(n))^2.$$

*This work was supported by the Office of Naval Research under Contract N00014-94-C-0128.

0-8194-1922-2/95/$6.00

Similarly, we can write the 2-norm of the transfer function matrix as

$$\|H\|_2 = \sum_{k=0}^{l_f-1} \sum_{i,j} |H_{ij}(k)|^2,$$

where $l_f \geq \max(l_e, l_s + l_d + l_w - 2)$ and

$$H_{ij}(k) = \frac{1}{\sqrt{l_f}} \sum_{n=0}^{l_f-1} h_{ij}(n) e^{2\pi\sqrt{-1}nk/l_f}.$$

Finite duration signals allow us to replace the standard integration over frequency with summation over k. By Parseval's theorem, we note that $\|\mathbf{H}\|_2 = \|H\|_2$, and the two will be used interchangeably in the remainder of this paper.

Problem QCQP can be considered a subset of the general nonlinearly constrained problem

$$\begin{array}{ll} \min\limits_{x \in \Re^n} & F(x) \\ \text{subject to} & c_i(x) = 0,\ i = 1, \ldots, , k'; \\ & c_i(x) \geq 0,\ i = k'+1, \ldots, k, \end{array} \tag{2}$$

restricted to quadratic $F(x)$ and quadratic inequality constraints. A whole body of literature exists for solving (2) under various different combinations of properties of $F(x)$ and $\{c_i(x)\}$ (e.g. refer to Gill, Murray & Wright[7]).

We prefer to couch our problem in the notation used in controller design theory, as our motivation stems from designing controllers for noise and vibration problems.

An extensive body of work, in both hardware and software, has been developed by the Applied Control and Signal Processing Group at SRI International. Several generations of hardware has been built to control multiple-input-multiple-output systems with multiple-actuators-multiple-sensors.

At least two existing controller design codes solve problems similar to (1). Boyd et al[2] developed a general control design compiler called *qdes* around a multiply linearly constrained quadratic optimization engine. SRI developed a FORTRAN software package, *SRI Weights Algorithm and Performance Simulator* (SWAPS),[9,10] around a single quadratically constrained quadratic optimization engine, which can also give sub-optimal results for an l_∞ constraint. These formulations use a matrix FIR representation for the Q-parameter (W). As an example from active noise control, Peterson, Weeks & Nowlin[11] use SWAPS and the latest generation control hardware to reduce the acoustic radiation from an air conditioner compressor.

Problem QCQP, treated in this paper, is a subset of the problems which can be solved by an SRI code called *MINCODE*, for Mixed-Norm Controller Design. MINCODE can design controllers where the objective function and constraints are any combination of H_∞, H_2 and l_∞ norms. A typical controller specification might require simultaneous constraints of out-of-band response, stability robustness, actuator authority, and sensitivity. Designing such a controller requires having an algorithm that can handle mixed-norm constraints, combinations of H_2, H_∞, l_1 and l_∞ norm constraints. This mixed-norm controller design problem can be posed as a convex optimization problem, and thus we can take advantage of a vast body of theory and algorithms. The particulars of the controller design problem are discussed in Section 3.

Our approach for MINCODE is to use the techniques of semi-definite programming, and in particular, a method which solves the primal and dual problems simultaneously. The approach and benefits are discussed in Section 4. We pose problem QCQP as a semi-definite program, which is far more general than the problem class we have. Yet we can then use the same code to solve many problems related to (1), such as different norms, other constraints, etc.

Vandenberghe & Boyd[12] thoroughly discuss the semi-definite programming approach, and the primal-dual potential reduction methods used here. However, they only consider the full matrix forms in the primal and dual problems. In the controller design problems we encounter, the number of variables and dimension of the problem can quickly number in the thousands. This calls for numerical tools to drastically cut down on the storage and operation count. These tools are presented in Section 5.

An example controller design problem, with random data, is presented in Section 6, where we look at tradeoffs between performance and constraint values.

Finally, in Section 7 we make several concluding remarks.

3 CONTROLLER DESIGN PROBLEM

The motivation for our controller design algorithm development stems from the problem of active noise or vibration control. The purpose of the active noise controller is to choose the optimal signals to send through a set of loudspeakers or actuators so as to cancel any unwanted noise or vibration in a designated quiet zone.

A typical setup is shown in figure 1. Noise sources or disturbances are sensed by an array of sensor microphones. A microprocessor implements W, the tap weights of FIR filters, and N, the neutralization filters, to create the "anti-noise" to send through an array of control actuators, or loudspeakers. Another array of sensors is situated in a designated quiet zone to monitor the effects of the disturbance. In a multichannel system any of the following can exist in multiples: the disturbances, the input or quiet zone sensors, and the actuators.

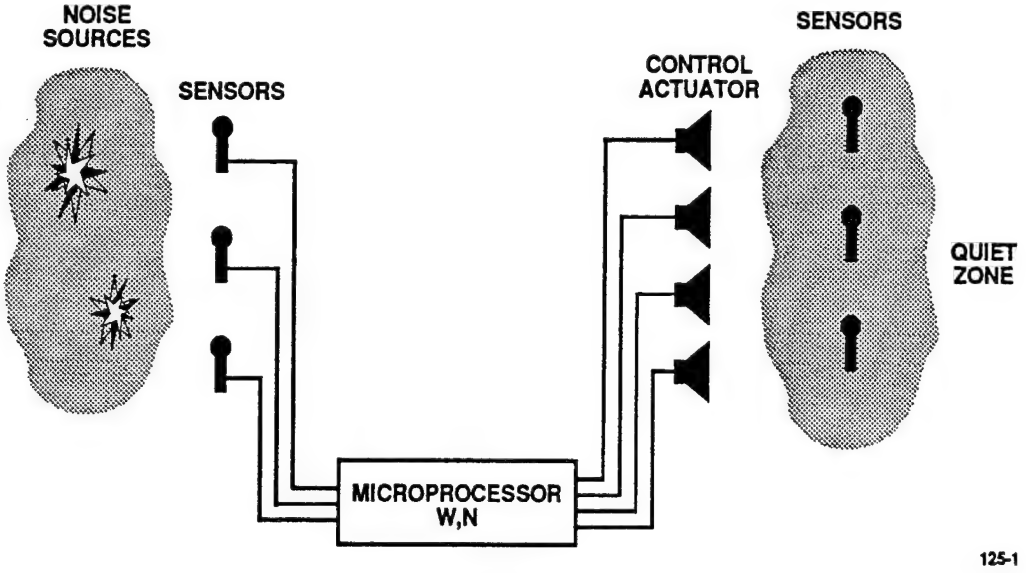

Figure 1: Multichannel active noise control system

Four pieces of information are required to solve for the filter weights: the transfer function from the disturbance to the microphones sensing the noise (denoted as S data for "sensors"), the transfer function from the loudspeakers to the error sensors (denoted as D data) and from the loudspeakers back to the noise-sensing microphones (denoted as C for "coupling" the anti-noise back through the system), and the transfer function from the disturbance to the error microphones (denoted as E data).

This data is collected as impulse responses. For the remainder of this paper, impulse responses will be denoted by bold face capital letters, and italic capital letters will denote transfer function matrices. It is assumed that the $(\mathbf{E}, \mathbf{S}, \mathbf{D})$ data is stable in that it does not grow, and eventually dies out. Computational requirements demand relatively short impulse responses (in the thousands of samples).

The data described above are displayed in Figure 2 which shows a general block diagram of a controlled plant. We assume that the plant P and the controller K are both linear, time-invariant systems. The plant consists

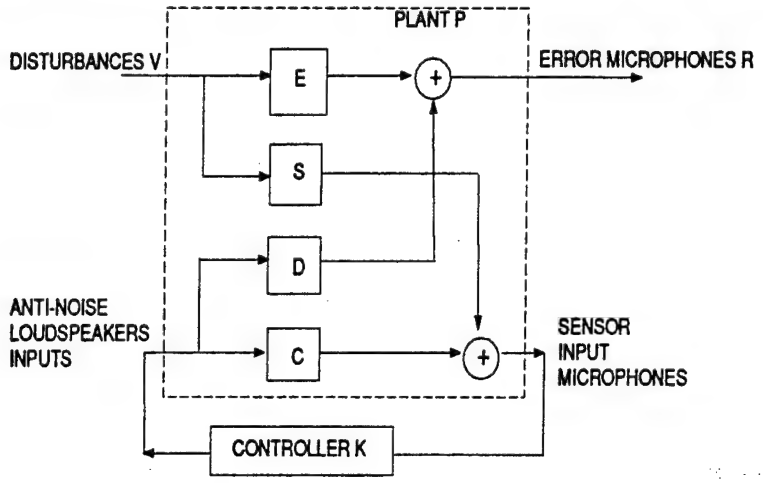

Figure 2: Block Diagram of Plant and Controller

of the E, S, D and C blocks. We assume that the controller consists of two blocks, the FIR filter weights W and the coupling neutralization filter N. Define H as the closed-loop transfer function (CLTF) which maps the disturbances V to the error microphones R, from Figure 2.

In general, the transfer function H is defined as

$$H = E + DK(I + CK)^{-1}S,$$

where the controller K is

$$K = W(I + NW)^{-1}.$$

However, when we use perfect neutralization, so that the neutralization filter N is chosen to be the opposite of the coupling, $N = -C$ then H simplifies to

$$H = E + DWS.$$

This approach is an example of $Q-$parametrization, or Youla parametrization, for stable plants. As W runs over the set of all stable, proper transfer matrices (of appropriate dimension), K runs over the set of all controllers that result in stable, realizable, closed-loop transfer matrices H(see Boyd & Barratt[1]). The Q-parametrization controller design methods determine the best possible performance for any stable, realizable, LTI controller given a set of constraints.

We denote

$$\mathbf{H}_i = \mathbf{E}_i + \mathbf{D}_i \circledast \mathbf{W} \circledast \mathbf{S}_i, \tag{3}$$

where $\mathbf{E}_i$, $\mathbf{S}_i$, and $\mathbf{D}_i$, are specified for each constraint. For $i = 0$, $\mathbf{H}_0$ denotes the nominal closed-loop disturbance-to-regulated-output impulse response. With this notation in place we can consider the design of a robust controller which satisfies many of the constraints listed above as the problem of

$$\begin{array}{ll} \text{minimize} & \|\mathbf{H}_0\|_{p_0} \\ \text{subject to} & \|\mathbf{H}_j\|_{p_j} \leq \alpha_j, \text{ for } j = 1, ..., n, \end{array} \tag{4}$$

where the norm p_j can be any of the ones listed above. The cost function in (4) is affine in $\mathbf{W}$, and thus convex, and the norm constraints are all convex, so the minimization in (4) is a convex optimization problem. The beauty of using convex optimization on the Q-parameter is that either the optimal controller will be designed, or else the user will know that no linear controller, designed by any means, will meet the given specifications.

The details of how (4) can be solved using convex optimization techniques are discussed in the next section.

4 SEMI-DEFINITE PROGRAMMING

The discussion in this section comes from Vandenberghe & Boyd,[12] and we refer the reader to their paper for more details on the theory and algorithms.

A semi-definite program (SDP) minimizes a linear function subject to an affine combination of symmetric matrices is semi-definite. In particular, the SDP can be written as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F(x) \geq 0 \end{array} \tag{5}$$

where

$$F(x) \equiv F_0 + \sum_{i=1}^{m} x_i F_i.$$

Equation (5) is also called the primal problem. It should be noted that we are couching problem QCQP as an SDP to take advantage of the algorithms and theory that exist in the literature. We can convert

$$\text{minimize } \|\mathbf{H}_0\|$$

to minimizing a linear function by

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|\mathbf{H}_0\| \leq t. \end{array} \tag{6}$$

For the remainder of this paper, all norms will be H_2 unless otherwise specified.

The norm of the closed-loop impulse response $\|\mathbf{H}_0\|$ can be written equivalently as $\|Xw + e\|_2$ where the matrix X is composed of the sensor (S) and actuator (D) data , the vector w is composed of the filter weights of the design variables, and e is a vector version of the disturbance data (E). Then the expression $\|\mathbf{H}_0\| \leq t$ can be written equivalently as a matrix inequality,

$$\begin{bmatrix} I & Xw+e \\ (Xw+e)^T & t \end{bmatrix} \geq 0. \tag{7}$$

The matrix depends affinely on the vector w and (7) can be expressed as

$$F(x) = F_0 + x_1 F_1 + ... + x_m F_m \geq 0, \tag{8}$$

where $w \in \Re^{m-1}$ and the variable x is the set $(t, w\}$. The matrices F_j in (8) are

$$F_0 = \begin{bmatrix} I & e \\ e^T & 0 \end{bmatrix}, \; F_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \; F_j = \begin{bmatrix} 0 & X_{j-1} \\ X_{j-1}^T & 0 \end{bmatrix}, \; j = 2, ..., m, \tag{9}$$

where X_j is the jth column of X. The coefficient c in the objective function $c^T x$ in (5) is $c = (1, 0, ..., 0)$.

It is straightforward to incorporate the (multiple) constraints from (4) into an SDP. Simply let the F_j matrices in (8) be block diagonal matrices with the first block corresponding to the appropriate F_j block of the objective

$\mathbf{H}_0$ and each successive block corresponding to the appropriate F_j block of the $\mathbf{H}_k$ constraint. The form of the F_j's shown in (9) are correct for constraining the H_2 norm of $\mathbf{H}_k$. Titterton[6] discusses minimizing and constraining the H_∞ norm of $\mathbf{H}_k$.

Our approach for solving (5) is to solve the primal and dual problems together, where the dual problem to (5) is

$$\begin{array}{ll} \text{maximize} & -\text{Trace}(F_0 Z) \\ \text{subject to} & \text{Trace}(F_i Z) = c_i, i = 1, ..., m \\ & Z \geq 0. \end{array} \tag{10}$$

Here $Z = Z^T \in \Re^{n \times n}$, and $Z \geq 0$ implies that Z is positive semi-definite. The primal-dual optimization problem is

$$\begin{array}{ll} \text{minimize} & c^T x + \text{Trace}(F_0 Z) \\ \text{subject to} & F(x) \geq 0, Z \geq 0 \\ & \text{Trace}(F_j Z) = c_j, j = 1, ..., m. \end{array} \tag{11}$$

The optimal value of this minimization is zero. The purpose is to minimize the primal and dual objective functions over all primal and dual feasible points. The reason for working with both the primal and dual problems together is that the dual information improves the next iteration along the primal direction, and vice-versa.

Define a potential function as

$$\varphi(x, Z) = (n + \nu\sqrt{n}) \log(\text{Trace}(F(x)Z)) - \log\det F(x) - \log\det Z - n \log n, \tag{12}$$

which combines, via a weighting factor ν, the duality gap, or sum of the primal and dual objective functions, with the deviation from centrality, or how far away we are from the minimizer of a barrier function on $F(x)$ and Z. The power in dealing with the potential function φ is that, if we start with a feasible x and Z, then all the iterates remain feasible and converge to the optimum in polynomial time. The general outline of the potential reduction algorithm is as follows:

Algorithm: Primal-Dual Potential Reduction (PDPR)
given strictly feasible x and Z
repeat
1. Find a suitable direction δx, and a suitable dual feasible direction δZ.
2. Find $p, q \in \Re$ that minimize $\varphi(x + p\delta x, Z + q\delta Z)$.
3. update: $x := x + p\delta x$ and $Z := Z + q\delta Z$.

until duality gap $\leq \epsilon$.

The suitable directions are found by Newton's method, and the minimizers along these directions can be carried out very efficiently, after some precomputations, using a guarded Newton method based on on work by Nesterov & Nemirovsky.[8] If the primal and dual updates remain feasible then there is a rigorous stopping criterion.

In using algorithm PDPR for control problems we must consider the computational cost of storage and floating point operations, because the filter tap weights can number in the thousands. Storing and working with the full matrices F_j and Z quickly proves unmanageable. We take full advantage of all inherent structure in the problem of minimizing and constraining the H_2 norm of $\mathbf{H}_j$, as discussed in the next section.

5 NUMERICAL TOOLS

In this section we present some of the numerical tools we use to take advantage of the inherent matrix structure in the H_2 controller design problem (4). The two labor-intensive portions of algorithm PDPR are finding a least

squares solution (for the primal Newton direction) and performing an eigenvalue decomposition in preparation for the plane search. These two parts are treated separately in the next two sections.

Before algorithm PDPR can be used, a strictly feasible x and Z must be found. A strictly feasible $x = (t, w)$ must satisfy $F(x) > 0$, which from (7) implies that we can set w to all zeros, and $t > \|e\|^2$.

We set the initial Z to a block diagonal identity matrix, with the number of blocks equal to one more than the number of constraints. Clearly this block concatenation of the identity matrix is positive definite. All we need to show is that $\text{Trace}(F_j Z) = c_j = (1, 0, ..., 0)$ for $j = 1, ..., m$. As can be seen from the form of the F_j's in (9), $F_j Z$ is zero for $j = 2, ..., m$ and $F_1 Z$ is one.

5.1 Find suitable search directions

Here we discuss how to find suitable search directions for the primal and dual problems. Suitability implies that we maintain feasibility.

5.1.1 Primal feasible

To compute the primal search direction we apply Newton's method to φ. As φ is composed of a concave plus convex term, we ignore the second derivative of the concave term when forming the Hessian. Using the notation in (7) we write $F(x)$ as

$$F(x) = \begin{bmatrix} I & Xw + e \\ (Xw + e)^T & t \end{bmatrix} = \begin{bmatrix} I & y \\ y^T & t \end{bmatrix}, \tag{13}$$

where we have set $y = Xw + e$ for notational ease, and $x = (t, w)$. Let $\gamma = \frac{1}{t - y^T y}$. Then the gradient is

$$\nabla \varphi = \begin{bmatrix} \nabla_t \varphi & \nabla_w \varphi \end{bmatrix} = \begin{bmatrix} -\gamma & 2\gamma X^T y \end{bmatrix}^T, \tag{14}$$

and the Hessian is

$$\nabla^2 \varphi \approx \begin{bmatrix} \gamma^2 & \left(-2\gamma^2 X^T y\right)^T \\ -2\gamma^2 X^T y & 2\gamma X^T X + \nabla_w \varphi \cdot \nabla_w \varphi^T \end{bmatrix}. \tag{15}$$

Due to the matrix product $X^T X$ and the vector outer product, it is too unwieldy to store the full Hessian matrix and solve for Hessian $\cdot\, \delta x =$ gradient directly. We use the iterative method of Conjugate Gradients (CG) instead as it only requires the action of the Hessian matrix on a given vector p. The vector outer product becomes simply the gradient scaled by the inner product $\nabla_w \varphi^T p$.

The $X^T X p$ portion of the Hessian-vector product in CG is accelerated using FFT's and Kronecker product identities. The details of the acceleration method can be found in Oetzel[9] and the application for the patent on SWAPS[7]). The main benefit is that calculating $X^T X p$ at each CG iteration requires $\mathcal{O}(n_s n_d l \log l)$ flops, where n_s is the number of sensors, n_d is the number of actuators and l is the number of FFT bins used, instead of $\mathcal{O}((n_s n_d l_w)^2)$, where l_w is the number of tap weights.

This use of Kronecker products and FFTs on circulant matrices is a fast way of computing the Hessian times p given any $\mathbf{E}, \mathbf{S}$ and $\mathbf{D}$. The same techniques can be used on computing the product of the Hessian of the constraints and direction p.

5.1.2 Dual feasible

The dual direction is given by

$$\delta Z = -\rho Z + F^{-1} - F^{-1}\delta F F^{-1}, \tag{16}$$

where $\delta F = \sum_{i=1}^{m} \delta x_i F_i$ for the Newton direction δx. If an exact Newton direction were found then this form of δZ would maintain feasibility in the dual problem. However, as we use an iterative method to solve for the Newton iteration, we need to force feasibility by including auxiliary variables to the original formulation of the problem.

The term $F^{-1} - F^{-1}\delta F F^{-1}$ in (16) can be written as

$$F^{-1} - F^{-1}\delta F F^{-1} = (I_z + gh^T), \tag{17}$$

where g, h are rank 2 matrices, computed anew at each Newton iteration and I_z is the identity matrix with a zero in the (n, n) entry instead of a one. Although the resulting matrix in (17) is symmetric, based on it's form it was more natural to choose the gh^T factorization rather than a symmetric factorization gg^T. Once the g and h^T matrices have been chosen, then all future factorizations, discussed below, will appear to result in nonsymmetric matrices, although that is not the case.

Since Z is updated by $Z = Z + p\delta Z$, the updated Z can be written as

$$Z := (1 - \rho)Z + (I_z + gh^T), \tag{18}$$

where ρ is the ratio of duality gap to $n + \nu\sqrt{n}$. Thus, from (18) we can see that at each iteration the matrix Z will be updated by a rank 4 matrix. Since Z begins as the block diagonal identity matrix, we can write a general form of Z as a diagonal concatenation of matrices of the form

$$Z_j = a_k I + A_k B_k^T, \tag{19}$$

where a_k is a scalar, and $A_k, B_k \in \Re^{n_j \times r}$ for small r. Here, the subscript k refers to the Newton iteration, and the subscript j refers to the objective ($j = 0$) or the jth constraint. These low-rank matrices and scalar are updated at each Newton iteration. Using the factorization in (19) we see that a block of δZ can be written as

$$\delta Z = \rho(I_z + gh^T) - (a_k I + A_k B_k^T). \tag{20}$$

5.2 Plane search

The plane search in step 2 of Algorithm PDPR requires finding the minimum of φ along the primal and dual search directions δx and δZ, respectively. Denote $F \equiv F(x)$, $\delta F \equiv \sum_{i=1}^{m} \delta x_i F_i$,

$$c_1 = \frac{c^T \delta x}{\text{Trace}(F(x)Z)}, \quad c_2 = \frac{\text{Trace}(F_0 \delta Z)}{\text{Trace}(F(x)Z)},$$

and $\mu_1, \ldots, \mu_n$ the generalized eigenvalues of $(\delta F, F)$ and $\nu_1, \ldots, \nu_n$ the generalized eigenvalues of $(\delta Z, Z)$. Then we safeguard the plane search by minimizing over p and q the following

$$\begin{array}{ll} \text{minimize} & (n + \nu\sqrt{n}) \log(1 + c_1 p + c_2 q) - \sum_{i=1}^{n} \log(1 + p\mu_i) - \sum_{i=1}^{n} \log(1 + q\nu_i) \\ \text{subject to} & p_{min} \le p \le p_{max}, q_{min} \le q \le q_{max}. \end{array}$$

It can be shown[5] that the objective function has a unique local minimum in the feasible rectangle which is the global minimum. Once the generalized eigenvalues are found, the plane search is cheap.

Finding the generalized eigenvalues of $(\delta Z, Z)$ means solving for λ in

$$\delta Z x = \lambda Z x.$$

Using the rank update forms in (19) and (20) and dropping the k subscript for notational ease gives

$$(\rho(I_x + gh^T) - (aI + AB^T))x = \lambda(aI + AB^T)x.$$

After some algebraic manipulation, and defining $\phi = (\lambda + 1 - \frac{\rho}{a})$ we have

$$(\rho g h^T - \frac{\rho}{a} AB^T - \rho N)x = \phi(aI + AB^T)x, \tag{21}$$

where N is the zero matrix with a one in the (n, n) entry. Define U_n to be the nth unit vector. Let P be the concatenation of ρg, $-\rho U_n$ and $-\frac{\rho}{a}A$, and Q be the concatenation of H^T, U_n and B^T. Then (21) becomes

$$(PQ^T)x = \phi(aI + AB^T)x.$$

However, this still involves computing an outer product and storing a size n matrix. So we pre-multiply by Q^T and post-multiply by P to arrive at the final form of the generalized eigenvalue problem in ϕ,

$$Q^T P Q^T P x = \phi(aQ^T P + Q^T A B^T P)x. \tag{22}$$

Before solving the generalized eigenvalue problem in (22) for ϕ and translate back to λ, we have to do one more step of massaging, because the matrix product $Q^T P Q^T P$ is rank deficient. Let

$$U \Sigma V^T = Q^T P Q^T P \tag{23}$$

be the singular value decomposition of the $((Q^T P)^2$ matrix, and let η be the rank of the singular value matrix Σ. We truncate U, Σ and V to contain just the singular values and vectors corresponding to the full rank part,

$$U_\eta = U(:, 1:\eta),\ V_\eta^T = V^T(1:\eta, :),\ \Sigma_\eta = \Sigma(1:\eta, 1:\eta).$$

Then we pre- and post-multiply both sides of (22) to get

$$V_\eta^T U_\eta \Sigma_\eta V_\eta^T U_\eta = \phi(aV_\eta^T(Q^T P + Q^T A B^T P)U_\eta)x. \tag{24}$$

Even though the dimension r of A and B^T grows with each Newton iteration, for large problems $r \ll n$. The svd in (23) and generalized eigenvalue problem in (24) is performed on an $r \times r$ matrix.

Finding the eigenvalues of $(F^{-1}\delta F)$ does not rely on low rank updates, but on algebraic manipulation. With $F(x)$ taking on the form in (13), then F^{-1} can be found analytically as

$$F^{-1} = \begin{bmatrix} I + \gamma y y^T & -\gamma y \\ -\gamma y & \gamma \end{bmatrix}. \tag{25}$$

The matrix δF has a sparse form:

$$\delta F = \begin{bmatrix} 0 & v \\ v^T & \tau \end{bmatrix}, \tag{26}$$

where $v \in \Re^n$ and τ is a scalar. Then $F^{-1}\delta F$ can be written as the product of two rank-2 matrices,

$$F^{-1}\delta F = \begin{bmatrix} -\gamma y & v \\ \gamma & 0 \end{bmatrix} \begin{bmatrix} v^T & y^T v - \tau \\ 0 & 1 \end{bmatrix}. \tag{27}$$

The nonzero eigenvalues of the outer product in (27) are the same as the eigenvalues of the inner product,

$$\begin{bmatrix} -\gamma y^T v - \gamma(y^T v - \tau) & \gamma \\ v^T v & 0 \end{bmatrix}.$$

6 RESULTS

As an example of a controller design which matches problem QCQP, we consider a plant with two input microphones, two actuators, two disturbances, and two error microphones. The data is randomly sampled from the normal distribution, the lengths of the S and D impulse responses is ten, the number of tap weights is four, and the length of E is 22 to match the convolution length of $\mathbf{D} \circledast \mathbf{W} \circledast \mathbf{S}$. We designed roughly 100 controllers on this plant, varying the constraint levels.

Define the performance at the error microphones to be $20 \log_{10}$ of the ratio of 2-norm of the closed-loop transfer matrix to the 2-norm of the open-loop transfer matrix. Negative numbers indicate an improvement of the closed-loop response compared to the open-loop response. We design a controller to minimize $\|\mathbf{H}_0\|_2$, that is, we minimize the RMS error due to white noise disturbance.

We investigate the effects of tightening the constraint levels α_j on performance. One constraint is on the effect of sensor noise to the output, defined by

$$\|\mathbf{D} \circledast \mathbf{W}\| \leq \alpha_1. \tag{28}$$

By decreasing α_1 we limit the bad contribution that sensor noise can make to the controller's ability to quiet the disturbances.

A simultaneous constraint is on the input to the actuators, given by

$$\|\mathbf{W} \circledast \mathbf{S}\| \leq \alpha_2. \tag{29}$$

If the disturbance is white noise then the H_2 norm is appropriate for the constraint. If we don't know characteristics of the disturbance then the H_∞ norm is appropriate in (29). The constraint in (29) is useful for those types of actuators which may be able to handle large peak input signals, but can't handle a large persistent input as they will overheat.

For this example we compute the exact W, W_{ex} for perfect cancellation. Using W_{ex}, we compute the values of the norms in (28) and (29), and create a sequence of ten constraint levels, each being .5 lower than the previous constraint level. All 100 combinations of these two constraint levels are run through MINCODE and the resulting performance calculated. Because the data is random, the performance values and constraint values have no intrinsic meaning, but can be treated as relative values. Figure 3 shows the mesh plot of the performance versus α_1 and α_2. The x-axis shows the various values of α_2 from (28), the y-axis shows the various values of α_1 from (29), and the z-axis shows the performance for each constraint combination. The tighter the constraint value the less performance is achieved.

Each data point in figure 3 represents a separate controller synthesis; that is, each point shows the optimal performance (in H2) for a single constraint combination. Given a priori specifications on the disturbance and sensor noise amplitudes and on the maximum RMS input into the actuator, we can find the optimal performance controller. As a whole, the plot shows the tradeoff between performance, $\|DW\|$, and $\|WS\|$. The designer can examine the effect on performance of larger actuators and/or less-noisy sensors, which enables sensible cost-for-performance tradeoffs.

7 CONCLUSIONS

In this paper we address how to solve a quadratically constrained quadratic program, with multiple constraints allowed. We pose this problem as a semi-definite programming problem because then the same code with minimal changes can also solve many other related problems, e.g. different norms, other constraints, etc.

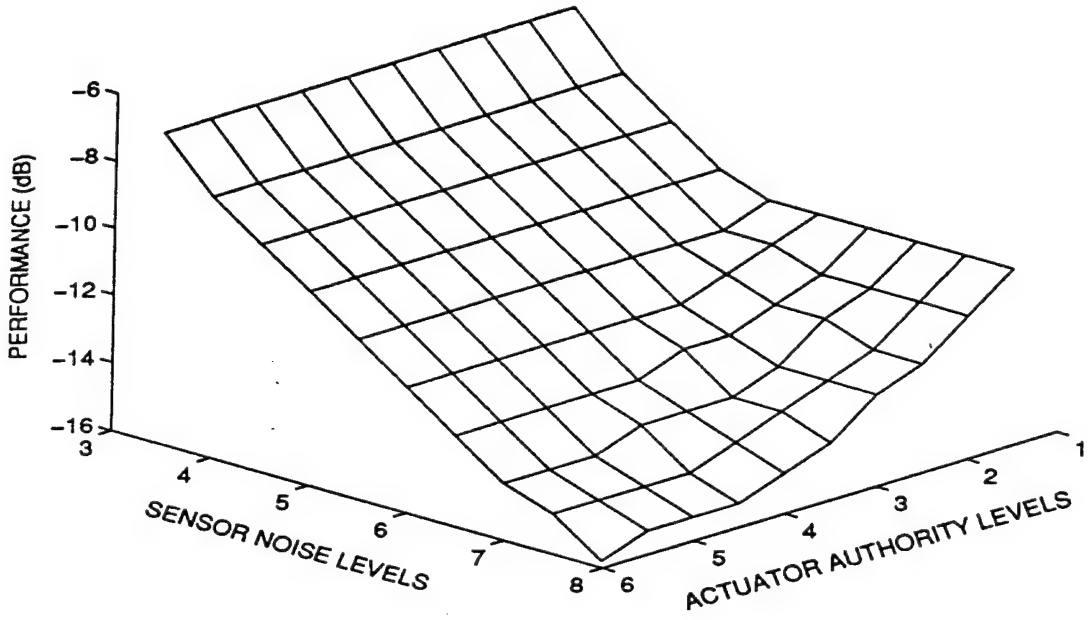

Figure 3: Tradeoffs of performance versus constraint levels

Our algorithm of choice is a primal-dual potential reduction method, in which we minimize the sum of the primal and dual objective functions simultaneously. Working on both problems together enables us to produce improved iterates for each problem.

The computer-intensive portions of the algorithm are in solving a least squares problem for the primal direction, and in computing two generalized eigenvalue problems in preparation for the plane search. We accelerate these two sections, and greatly reduce the storage needed, by using Kronecker product identities, FFT's on extended matrices, and updated matrix factorizations.

We present an example of examining the tradeoff between performance of a controller and two constraints. One constraint represents a physical limit on actuator authority, and the other constraint represents how we can limit the effects of sensor noise on the error microphone output.

8 ACKNOWLEDGEMENTS

The authors are grateful to Stephen Boyd for invaluable help with the theory and practical considerations regarding semi-definite programming, and for his editorial comments for this paper.

9 REFERENCES

[1] Stephen Boyd and Craig Barratt. *Linear Controller Design: Limits of Performance.* Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[2] Stephen P. Boyd, Venkataramanan Balakrishnan, Craig H. Barratt, Nasser M .Khraishi, Xiaoming Li, Kavid G. Meyer, and Stephen A. Norman. A new cad method and associated architectures for linear controllers. *IEEE Transactions on Automatic Control.*, 33(3):268–283, 1988.

[3] Tony F. Chan and Donald W. Cooley Julia A. Olkin. Solving quadratically constrained least squares using black box solvers. *BIT*, 32:481–495, 1992.

[4] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization.* Academic Press, Harcourt Brace and Company, 1981.

[5] M. Iri and H. Imai. A multiplicative barrier function method for linear programming. *Algorithmica*, 1:455–482, 1986.

[6] Paul J. Titterton Jr. Practical multi-constraint h^∞ controller synthesis from time-domain data. Submitted to *International Journal of Robust and Nonlinear Control.*, 1995.

[7] Martinez, Cooley, Oetzel, Woodworth, Olkin, Hessing, Peterson, Steinberger, and Mason. Method of calculating filter weights for compression waves cancellation systems. Patent P3161, pending, SRI International, 1991.

[8] Y. Nesterov and A. Nemirovsky. *Interior Point Polynomial Algorithms in Convex Programming.* Society for Industrial and Applied Mathematics, 1994.

[9] Kenneth G. Oetzel. A fast algorithm for calculating multi-channel wiener filters. Internal report of the System Technology Division, SRI International, 1989.

[10] Julia Olkin, Michele Freed, Chris Gellrich, and Patricia Jungers. Swaps user's manual. Internal report of the System Technology Division, SRI International, 1994.

[11] D. Kent Peterson, William A. Weeks, and William C. Nowlin. Active control of complex noise problems using a broadband, multichannel controller. *Noise-Con 94*, pages 315–320, 1994.

[12] Lieven Vandenberghe and Stephen Boyd. Semi-definite programming. To appear in *SIAM Review.*, 1995.

Journal of VLSI Signal Processing 14, 57–66 (1996)
© 1996 Kluwer Academic Publishers. Manufactured in The Netherlands.

Using Semi-Definite Programming for Multi-Constrained H_2 Controller Design in Active Noise and Vibration Control*

JULIA A. OLKIN AND PAUL J. TITTERTON, JR.
Applied Control and Signal Processing Group, SRI International, Menlo Park, CA 94025

Received January 31, 1996; Revised April 17, 1996

Abstract. We consider the practical design of linear controllers to meet a given set of H_2 specifications. The Q-parametrization reduces the problem to a quadratic minimization subject to multiple quadratic constraints, which we solve using semi-definite programming (SDP) methods. Each SDP iteration requires calculating a primal and dual search direction and minimizing the cost function along the plane defined by these search directions. The primal direction requires solving a least squares problem whose normal equation matrix is composed of a block-Toeplitz portion plus other structured matrices. We make use of Kronecker products and FFT's to greatly reduce the calculation. The dual search direction and plane search are accelerated by low-rank representations of the SDP structured matrices.

As an example, we design controllers which explore the optimal tradeoff between in-band residual and out-of-band enhancement of acoustic radiation from a (mathematically modeled) submerged spherical shell, while simultaneously constraining two sensitivity measures. For this example we show that significant reduction in out-of-band enhancement is possible with only minor in-band penalties.

1. Introduction

We design controllers to minimize the H_2 norm of a closed-loop transfer function subject to simultaneous multiple H_2 norm constraints. The controllers are assumed to be discrete-time, linear time-invariant (LTI), and the minimization function is a norm on the map from disturbances to regulated outputs. Using Q-parametrization, this problem can be formulated as

$$\begin{aligned} \min_{\mathbf{W}} \quad & \|\mathbf{E}_0 + \mathbf{D}_0 \circledast \mathbf{W} \circledast \mathbf{S}_0\|_2 \\ \text{subject to} \quad & \|\mathbf{E}_j + \mathbf{D}_j \circledast \mathbf{W} \circledast \mathbf{S}_j\|_2 \le \alpha_j, \\ & \qquad \text{for } j = 1, \ldots, k, \end{aligned} \tag{1}$$

referred to as Problem QCQP, for quadratically constrained quadratic programming. The data $(\mathbf{E}_j, \mathbf{S}_j, \mathbf{D}_j)$ are finite-duration matrix functions of discrete time, $\mathbf{W}$ is the design variable, and $\circledast$ is the discrete time convolution operator[1].

*This work was supported by the Office of Naval Research under Contract N00014-94-C-0128.

The work presented here comprises part of our larger project, which is to solve the more general form of (1),

$$\begin{aligned} \min_{\mathbf{W}} \quad & \|\mathbf{E}_0 + \mathbf{D}_0 \circledast \mathbf{W} \circledast \mathbf{S}_0\|_{p_0} \\ \text{subject to} \quad & \|\mathbf{E}_j + \mathbf{D}_j \circledast \mathbf{W} \circledast \mathbf{S}_j\|_{p_j} \le \alpha_j, \\ & \qquad \text{for } j = 1, \ldots, k, \end{aligned} \tag{2}$$

where the norms can be any combination of H_2, H_∞, l_∞ and l_1. In this paper we are restricting ourselves to the H_2 norm. Rather than create a separate package which deals only with Problem QCQP, and could thus be even further fine-tuned, we prefer to have the solution to (1) treated as a special case of (2) so that it is easy and straightforward to change the norms p_j to any of the norms mentioned above.

The ability to design controllers which must satisfy multiple specifications simultaneously is powerful. For example, out-of-band response, stability robustness, actuator authority, and sensitivity could all be constrained while optimizing the in-band performance. We have developed a software package *MINCODE*,

for *Mi*xed-*N*orm *Co*ntroller *De*sign, which solves (2). While this paper concentrates on our H_2 work, Titterton [1] discusses our work on minimizing and constraining the H_∞ norm of $\mathbf{H}_k$.

The H_2 norm can be computed from its impulse response matrix by

$$\|\mathbf{H}\|_2 = \sum_n \sum_{i,j} (h_{ij}(n))^2.$$

Similarly, we can compute the H_2 norm from the transfer function matrix as

$$\|H\|_2 = \sum_{k=0}^{l_f-1} \sum_{i,j} |H_{ij}(k)|^2,$$

where $l_f \geq \text{length}(\mathbf{H})$ and

$$H_{ij}(k) = \frac{1}{\sqrt{l_f}} \sum_{n=0}^{l_f-1} h_{ij}(n) e^{2\pi\sqrt{-1}nk/l_f}.$$

Finite duration signals allow us to replace the standard integration over frequency with summation over k. This norm can also be viewed as the RMS value of the multichannel output signal when the inputs are driven by independent white noise. The H_2 norm is used primarily to minimize the RMS response of the system for white or colored noise disturbances. For the remainder of this paper, impulse responses will be denoted by bold face capital letters, and italic capital letters will denote transfer function matrices.

Our contribution is a practical method to treat multiple H_2 controller design specifications simultaneously. Using convex optimization methodology on the Q-parametrization of the controller enables us to design the optimal discrete-time LTI controller that minimizes the $\mathbf{H}_2$ norm of a closed-loop transfer function and satisfies the multiple constraints. The convex optimization algorithm requires starting with a feasible point, computed by using the same algorithm on a modification of the constrained minimization problem. If no such starting feasible point can be found then no discrete-time LTI controller exists satisfying all the H_2 constraints.

Convex optimization problems can be solved expediently by using tailored algorithms based on structure in the problem. Our approach is to treat Problem QCQP as a semi-definite program (SDP), that is, the family of minimization problems in which the cost function is linear and the constraints are affine combinations of symmetric matrices being semi-definite (explained in more detail in Section 3.1). Our algorithm of choice, minimizing a primal-dual potential function, provides a rigorous stopping criterion so it is clear when the algorithm has converged, has faster convergence over the primal methods, and an iterative procedure for finding a starting feasible point that determines conclusively if the constraints are infeasible.

Our final contribution is to greatly reduce the number of floating point computations and storage required by the SDP algorithm. In its general form, each primal/dual iteration requires many large internal matrices. By using Kronecker product identities, FFT's on matrices which have been extended to circulant matrices, and low-rank representations of some of the internal matrices, we cut down considerably on the storage and flops. This computational and storage reduction allows us to design controllers for larger plants, that is, the $\mathbf{E}$, $\mathbf{S}$ and $\mathbf{D}$ data can be large in both length and in number of inputs and outputs, and the design variable W can be long enough to enable a good solution to the minimization problem.

We have treated the structure in Problem QCQP by SDP methods. However, the literature contains abundant studies of how to solve the general nonlinearly constrained problem, that is, minimizing a general nonlinear function subject to both equality and inequality constraints. Gill, Murray & Wright [2], for example, discuss the best methods to use under various different combinations of properties of the objective function and the constraints. Due to its properties, Problem QCQP could be solved by successive quadratic programming techniques.

Our software package allows for multiple H_2 constraints. For multiple linear constraints, Boyd et al. [3] have developed a general control design compiler called *qdes* around a multiply linearly constrained quadratic optimization engine. For H_2 minimization with one constraint allowed, SRI has developed a FORTRAN software package, *SRI Weights Algorithm and Performance Simulation* [4, 5].

MATLAB[2] contains the LMI control toolbox [6] which can solve Problem QCQP but only for small problem sizes because matrices involved in the formulation quickly grow with problem size.

Software packages which use state-space solutions could be used by massaging Problem QCQP into an unconstrained problem by minimizing a weighted sum of the objective and constraints. Then many state-space

solutions could be found by varying the values of the weighting parameters and creating a trade-off curve between weighting values and minimization value. However, each new constraint adds another dimension to search over, and with even 2 or 3 constraints the search becomes unwieldy.

The remainder of the paper is organized as follows. The particulars of the controller design problem and how it can be posed as Problem QCQP are discussed in Section 2. In Section 3 we show how to write Problem QCQP as a SDP, and discuss the numerical tools we have developed to drastically cut down on the storage and operation count.

In this paper we demonstrate our techniques on an active noise control problem. Information from two arrays of sensors and an array of actuators enables us to formulate a controller synthesis problem. The controller is used to reduce the noise or vibration in a prescribed physical area. We discuss this application in more detail in Section 4. We include an example active noise controller design problem of minimizing the sound radiating from a (mathematically modeled) submerged spherical shell, where we investigate tradeoffs between in-band residuals and out-of-band enhancement.

Finally, in Section 5 we make several concluding remarks, summarizing the contributions of the paper.

2. Controller Design Problem

A block diagram of a controlled plant is displayed in Fig. 1.

We describe the plant by four finite-length impulse response matrices: **E** is the disturbance-to-regulated-output response, **S** is the disturbance-to-control-sensor response, **D** is the control-actuator-to-regulated-output response, and **C** is the control-actuator-to-control-sensor response. The number of disturbance inputs is n_t, the number of control sensors is n_s, the number of control actuators is n_d, and the number of regulated outputs is n_r. The data **E** is dimension (n_r, n_t, l_e), **D** is dimension (n_r, n_d, l_d), **S** is dimension (n_s, n_t, l_s), **C** is dimension (n_s, n_d, l_c), and the design variable **W** is dimension (n_d, n_s, l_w) where l_j, $j = \{e, d, s, c, w\}$ represents the length of the respective FIR matrices. Note that the numbers of disturbances and regulated outputs will not necessarily be the same for the minimization criteria and the constraints.

We assume that the plant dynamics are fully described by its FIR's. Computational requirements

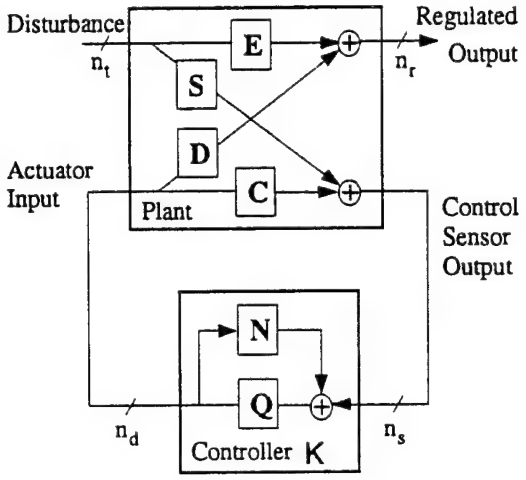

Figure 1. Plant and controller topology and notation.

demand relatively short impulse responses (in the thousands of samples). Further assumptions are that both the plant P and controller K are LTI, and that the controller consists of two blocks, the Q-parameter W, and the coupling neutralization filter N. Define H as the closed-loop transfer function which maps the disturbances to the regulated outputs. Setting $N = -C$ simplifies H to

$$H = E + DWS.$$

This approach is the simplest example of Q-parametrization, or Youla parametrization, for stable plants. As W runs over the set of all stable, proper transfer matrices (of appropriate dimension), K runs over the set of all controllers that result in stable, realizable, closed-loop transfer matrices H [7].

We denote

$$\mathbf{H}_i = \mathbf{E}_i + \mathbf{D}_i \circledast \mathbf{W} \circledast \mathbf{S}_i, \tag{3}$$

where $\mathbf{E}_i$, $\mathbf{S}_i$, and $\mathbf{D}_i$, are specified for each constraint. For $i = 0$, $\mathbf{H}_0$ denotes the nominal closed-loop disturbance-to-regulated-output impulse response. With this notation in place we can rewrite Problem QCQP as

$$\begin{array}{ll} \min\limits_{\mathbf{W}} & \|\mathbf{H}_0\|_2 \\ \text{subject to} & \|\mathbf{H}_j\|_2 \le \alpha_j, \quad \text{for } j = 1, \ldots, n. \end{array} \tag{4}$$

The cost function and constraints are convex so the minimization in (4) is a convex optimization problem.

Convex optimization techniques either finds the unique global minimizer or conclusively determines that no feasible point exists. In the case of controller design, the optimization technique either finds a controller which minimizes the H_2 norm of the CLTF $\mathbf{H}$ or determines that no realizable controller exists which meets all the constraints.

3. Numerical Tools

In this section we translate Problem QCQP into an SDP, and we show how to exploit the problem structure to speed its solution. For more details regarding SDP than we provide here, we refer the reader to Vandenberghe and Boyd [8], who thoroughly discuss the semi-definite programming approach, and the primal-dual potential reduction methods we use.

Vandenberghe and Boyd have developed algorithms geared to work particularly on SDP, but their software package uses many large internal matrices. For these algorithms to be viable for many controller design problems, we need to take full advantage of all inherent structure in the problem, and these numerical tools are shown here.

3.1. *Writing Problem QCQP as a Semi-Definite Program*

A semi-definite program (SDP) minimizes a linear function subject to an affine combination of symmetric matrices being semi-definite. In particular, the SDP can be written as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F(x) \geq 0 \end{array} \tag{5}$$

where

$$F(x) \equiv F_0 + \sum_{i=1}^{m} x_i F_i, \quad F_i \text{ symmetric.}$$

We can convert the objective function in (4)

$$\text{minimize } \|\mathbf{H}_0\|$$

to minimizing a linear function by adding a new variable t,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|\mathbf{H}_0\| \leq t. \end{array} \tag{6}$$

The H_2 norm of a closed-loop impulse response $\|\mathbf{H}\|$ can be written equivalently as $\|Xw + e\|_2$ where the matrix X is composed of the S and D data , w is a vector version of the design variable W, and e is a vector version of the E data. Then the expression $\|\mathbf{H}\|^2 \leq t$ can be written equivalently as the matrix inequality

$$\begin{bmatrix} I & Xw+e \\ (Xw+e)^T & t \end{bmatrix} \geq 0, \tag{7}$$

where the unknown variable x in (5) is now $\{t, w\}$. The matrix depends affinely on the variables $\{t, w\}$ and (7) can be expressed as

$$F(x) = F_0 + x_1 F_1 + \cdots + x_m F_m \geq 0. \tag{8}$$

The matrices F_j in (8) are

$$F_0 = \begin{bmatrix} I & e \\ e^T & 0 \end{bmatrix}, \quad F_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$
$$F_j = \begin{bmatrix} 0 & X_{j-1} \\ X_{j-1}^T & 0 \end{bmatrix}, \quad j = 2, \ldots, m, \tag{9}$$

where X_j is the jth column of X. The coefficient c in the objective function $c^T x$ in (5) becomes $c = (1, 0, \ldots, 0)$.

It is straightforward to incorporate the (multiple) constraints from (4) into an SDP. Simply let the F_j matrices in (8) be block diagonal matrices with the first block corresponding to the appropriate F_j block of the objective $\mathbf{H}_0$ and each successive block corresponding to the appropriate F_j block of the $\mathbf{H}_k$ constraint. The form of the F_j's shown in (9) are correct for constraining the H_2 norm of $\mathbf{H}_k$.

We present a general outline of the primal-dual algorithm used to solve the SDP.

Algorithm:
given strictly feasible primal and dual points
repeat until stopping criterion is small enough

1. Solve for the primal search direction
2. Compute dual search direction
3. Plane search: find the minimum point along the plane defined by primal and dual directions
   - Primal eigenvalue decomposition
   - Dual eigenvalue decomposition

The suitable directions are found by Newton's method, and the minimizers along these directions can

be carried out very efficiently, after some precomputations, using a guarded Newton method based on work by Nesterov and Nemirovsky [9].

The storage and manipulation of the matrices in the steps of the algorithm can quickly become untenable as the data and design variables grow in number and size. How we deal with this growth is discussed in the next section.

3.2. *Using Numerical Tools*

The two labor-intensive portions of the algorithm are finding a least squares solution (for the primal Newton direction) and performing an eigenvalue decomposition in preparation for the plane search. We treat these two parts separately in the following two sections.

To start the algorithm, a strictly feasible primal and dual point must be given. If all the $\mathbf{E}_j$, $j = 1, \ldots, k$ in (1) are zero then setting the primal variable w to all zeros, and $t > \|e\|^2$ gives a strictly feasible primal point. However, if any of the $\mathbf{E}_j$ are nonzero then a preliminary step is required. For one constraint, start searching for the solution to $\min\|\mathbf{H}_1\|$ and stop as soon as a $\mathbf{W}$ is found such that $\|\mathbf{H}_1\| \leq \alpha_1$. This $\mathbf{W}$ is the primal starting variable. If there is more than one such constraint, e.g., the second constraint, then the $\mathbf{W}$ just found is used as a starting value in the constrained problem $\min\|\mathbf{H}_2\|$ subject to $\|\mathbf{H}_1\| \leq \alpha_1$. The algorithm can be stopped as soon as a new $\mathbf{W}$ is found such that $\|\mathbf{H}_2\| \leq \alpha_2$. Otherwise, if the solution to this constrained problem is found and it does not satisfy $\|\mathbf{H}_2\| \leq \alpha_2$ then no feasible point exists. This procedure of solving increasingly larger constrained minimization problems continues until all constraints have been accounted for. Setting Z, the dual variable, to a block diagonal identity matrix, with the number of blocks equal to one more than the number of constraints, gives a strictly feasible dual point.

Primal Feasible Search Direction. To compute the primal search direction we apply a modified Newton's method to the potential function

$$\varphi(x, Z) = (n + \nu\sqrt{n}) \log(\text{Trace}(F(x)Z)) - \log\det F(x) - \log\det Z - n\log n, \quad (10)$$

which combines, via a weighting factor ν, the duality gap, or sum of the primal and dual objective functions, with the deviation from centrality, or how far away we are from the minimizer of a barrier function on $F(x)$ and Z. The power in dealing with the potential function φ is that, if we start with a feasible x and Z, then all the iterates remain feasible and converge to the optimum in polynomial time. As φ is composed of a concave plus convex term, we ignore the second derivative of the concave term when forming the Hessian. Using the notation in (7) we write $F(x)$ as

$$F(x) = \begin{bmatrix} I & Xw + e \\ (Xw + e)^T & t \end{bmatrix} = \begin{bmatrix} I & y \\ y^T & t \end{bmatrix}, \quad (11)$$

where we have set $y = Xw + e$ for notational ease. Let $\gamma = \frac{1}{t - y^T y}$. Then the gradient is

$$\nabla\varphi = \begin{bmatrix} \nabla_t\varphi & \nabla_w\varphi \end{bmatrix}^T = \begin{bmatrix} -\gamma & 2\gamma y^T X \end{bmatrix}^T, \quad (12)$$

and the Hessian is

$$\nabla^2\varphi \approx \begin{bmatrix} \gamma^2 & \left(-2\gamma^2 X^T y\right)^T \\ -2\gamma^2 X^T y & 2\gamma X^T X + \nabla_w\varphi \cdot \nabla_w\varphi^T \end{bmatrix}. \quad (13)$$

Due to the presence of the matrix product $X^T X$ and the vector outer product, it is too unwieldy to store the full Hessian matrix and solve for the Newton direction directly. We use the iterative method of Conjugate Gradients (CG) instead as it only requires the action of the Hessian matrix on a given vector p. The vector outer product becomes simply the gradient scaled by the inner product $\nabla_w\varphi^T p$.

The $X^T Xp$ portion of the Hessian-vector product in CG is accelerated using FFT's and Kronecker product identities. The details of the acceleration method can be found in Oetzel [10]. The main benefit is that calculating $X^T Xp$ at each CG iteration requires $\mathcal{O}((n_s n_d)^{\frac{3}{2}} l \log l)$ flops, where l is the number of FFT bins used, instead of $\mathcal{O}((n_s n_d l_w)^2)$, where l_w is the number of tap weights in the Q-parameter.

This use of Kronecker products and FFTs is a fast way of computing the Hessian times p given any $\mathbf{E}$, $\mathbf{S}$ and $\mathbf{D}$. The same techniques can be used on computing the product of the Hessian of the constraints and direction p.

Dual Feasible Search Direction. The dual search direction is given by

$$\delta Z = -Z + \rho(F(x)^{-1} - F(x)^{-1}\delta F F(x)^{-1}), \quad (14)$$

where $\delta F = \sum_{i=1}^{m} \delta x_i F_i$ for the Newton direction δx, and ρ is a scalar.

The term $F^{-1} - F^{-1}\delta F F^{-1}$ in (14) can be written as

$$F^{-1} - F^{-1}\delta F F^{-1} = (I + gNg^T), \qquad (15)$$

where g is a rank 3 matrix and N is a weighting matrix, both re-computed at each Newton iteration.

Since Z is updated by $Z = Z + p\delta Z$, for a scalar p, the updated Z can be written as

$$Z := (1 - p)Z + p\rho(I + gNg^T). \qquad (16)$$

Thus, from (16) we can see that at each iteration the matrix Z will be updated by a rank 3 modification of the identity. Since Z begins as the block diagonal identity matrix, we can write a general form of Z as a diagonal concatenation of matrices of the form

$$Z = a_k I + A_k M_k A_k^T, \qquad (17)$$

where a_k is a scalar, $A_k \in \Re^{n_j \times r}$ for small r, and M is a weighting matrix. Here, the subscript k refers to the Newton iteration, and the subscript j refers to the objective ($j = 0$) or the jth constraint. The low-rank matrix and scalar are updated at each Newton iteration. Using the factorization in (17) we see that a block of δZ can be written as

$$\delta Z = \rho(I + gNg^T) - \left(a_k I + A_k M_k A_k^T\right). \qquad (18)$$

Plane Search. The plane search along the primal and dual directions, δx and δZ, respectively, can be computed cheaply once the generalized eigenvalues $\mu_1, \ldots, \mu_n$ of $(\delta F, F(x))$ and generalized eigenvalues $\nu_1, \ldots, \nu_n$ of $(\delta Z, Z)$ are found. It can be shown [11] that the plane search can be posed as finding the unique local minimum in a specific feasible rectangle using a safeguarded Newton method. Once the generalized eigenvalues are found, the plane search is cheap.

Finding the generalized eigenvalues of $(\delta Z, Z)$ means solving for λ in

$$\delta Z x = \lambda Z x.$$

Using the rank update forms in (17) and (18) and dropping the k subscript for notational ease gives

$$(\rho(I + gNg^T) - (aI + AMA^T))x = \lambda(aI + AMA^T)x.$$

After some algebraic manipulation, and defining $\phi = (\lambda + 1 - \frac{\rho}{a})$ we have

$$\left(\rho gNg^T - \frac{\rho}{a}AMA^T\right)x = \phi(aI + AMA^T)x. \qquad (19)$$

Let P be the concatenation of g and A, and the block diagonal matrix R be $\mathrm{diag}(\rho N, -\frac{\rho}{a}M)$. Then (19) becomes

$$(PRP^T)x = \phi(aI + AMA^T)x. \qquad (20)$$

However, this still involves computing an outer product and storing a rank-deficient size n matrix. We deal with both the size and rank-deficiency problem with one pre- and post-multiplication. Compute the singular value decomposition of P as

$$U\Sigma V^T = P \qquad (21)$$

and let η be the rank of the singular value matrix Σ. We truncate U, Σ and V to contain just the singular values and vectors corresponding to the full rank part,

$$\begin{aligned} U_\eta &= U(:, 1:\eta), \\ V_\eta^T &= V(1:\eta, :)^T, \\ \Sigma_\eta &= \Sigma(1:\eta, 1:\eta). \end{aligned}$$

Then we pre- and post-multiply both sides of (20) to get

$$\left(U_\eta^T P\right)R(P^T U_\eta)x = \phi\left(aI + \left(U_\eta^T A\right)M(A^T U_\eta)\right)x. \qquad (22)$$

The SVD in (21) and generalized eigenvalue problem in (22) are performed on an $n \times r$ matrix. Even though the dimension r of A grows with each Newton iteration, for large problems $r \ll n$ and thus the SVD and eigenvalue decompositions are very fast.

Finding the eigenvalues of $(F^{-1}\delta F)$ does not rely on low rank updates, but on algebraic manipulation. With $F(x)$ taking on the form in (11), then F^{-1} can be found analytically as

$$F^{-1} = \begin{bmatrix} I + \gamma yy^T & -\gamma y \\ -\gamma y & \gamma \end{bmatrix}. \qquad (23)$$

The matrix δF has a sparse form:

$$\delta F = \begin{bmatrix} 0 & v \\ v^T & \tau \end{bmatrix}, \qquad (24)$$

where $v \in \Re^n$ and τ is a scalar. Then $F^{-1}\delta F$ can be written as the product of two rank-2 matrices,

$$F^{-1}\delta F = \begin{bmatrix} -\gamma y & v \\ \gamma & 0 \end{bmatrix} \begin{bmatrix} v^T & y^T v - \tau \\ 0 & 1 \end{bmatrix}. \quad (25)$$

The nonzero eigenvalues of the outer product in (25) are the same as the eigenvalues of the inner product,

$$\begin{bmatrix} -\gamma y^T v - \gamma(y^T v - \tau) & \gamma \\ v^T v & 0 \end{bmatrix}.$$

4. Optimal H_2 Tradeoff Study for Radiated Sound from Submerged Spherical Shell

This section focuses our discussion of multi-constraint H_2 controller synthesis on the active noise/vibration control problem. We first describe the mathematical model of the submerged spherical shell, and then show the results of a control design study in which we trade off out-of-band radiation enhancement for in-band residual while constraining two different sensitivity measures.

A typical noise or vibration controller setup is shown in Fig. 2. Noise sources or disturbances pass through an array of control sensor. A microprocessor incorporates W, the Q-parameter, and N, the neutralization path, as FIR filters, to create the anti-noise which is sent through an array of control actuators. The hardware design to implement the filters is straightforward (see Weeks and Curless [12]). Another array of sensors is situated in a designated quiet zone to monitor the effects of the disturbance. In a multichannel system any of the following can exist in multiples: the disturbances, the control sensors, the actuators, and the regulated output.

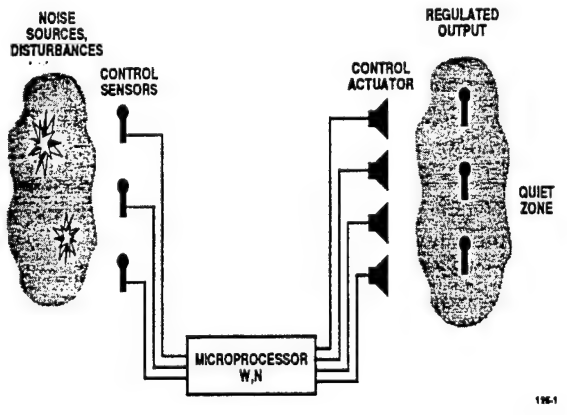

Figure 2. Multichannel active noise control system.

Table 1. Element locations (α, β) in degrees.

	Disturbance	Control sensor	Control actuator	Far field pressure
1	(0,0)	(17,175)	(35,0)	(180,0)
2	(5,0)	(15,280)	(30,0)	(175,0)

In this design example, we control the acoustic radiation from a submerged spherical shell. An analytical model is used to simulate measured (sampled) impulse responses. The plant has two disturbance sources, two control sensors, two control actuators, and two radiated field (regulated output) sensors.

4.1. *System Description*

With reference to Fig. 3, the disturbance is modeled as two internal normal-force sources located near the sphere's north pole ($+z$). Two control sensors and two control actuators are placed on the shell in the region adjacent to the disturbance. Finally, two far-field pressure sensors are arrayed with 5° spacing off the sphere's south pole. Table 1 gives the exact element locations, where α is degrees latitude, measured from the $+z$ axis, and β is degrees longitude, measured from the $+x$ axis (not shown in Fig. 3).

Again, with reference to Fig. 3, we define the speed of sound in the exterior fluid, $c = 1480$ m/s; the fluid's density, $\rho_0 = 1000$ kg/m^3; the sphere's radius, $a = 5$ m; thickness, $h = 0.15$ m; Young's modulus, $E = 19.6 \times 10^{10} \times (1 + 0.08i)$ N/m^2; Poisson's ratio, $\nu = 0.3$; and material density, $\rho_s = 7668.7$ kg/m^3. Complex Young's modulus is included to represent material damping in the shell. The regulated outputs are the far-field radiated pressure referenced to 100 m. We assume that the control sensors measure normal (outward) velocity and that the actuators provide a normal force. The dynamics of the disturbance sources are assumed to be that of an eighth-order Butterworth band-pass filter with low- and high-frequency cutoff at 50 and 200 Hz. The control actuator dynamics are similar, with cutoff at 50 and 300 Hz. The actuator and disturbance dynamics also serve as anti-alias filters.

We use the standard bending shell approximation, as presented in Junger and Feit [13], for the spherical shell, coupled to an exact solution for the infinite-extent fluid. The normal-force-to-normal-velocity transfer

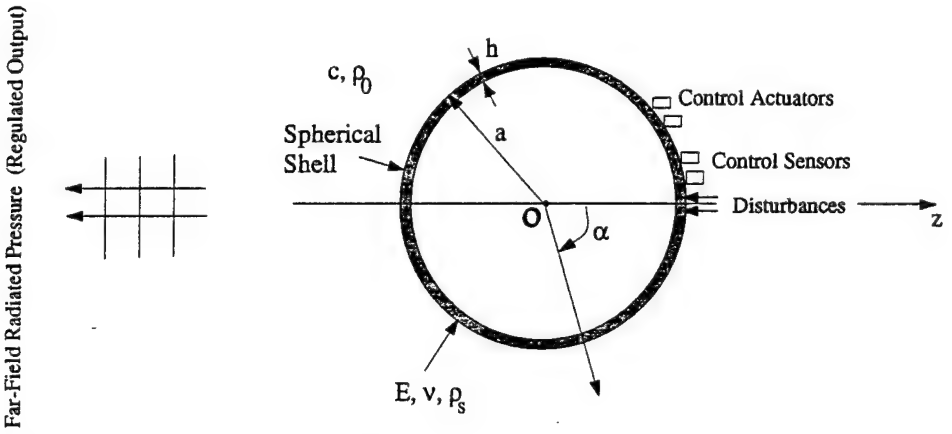

Figure 3. Physical configuration for the controller problem.

function is given by Junger and Feit Eq. (9.13), and the normal-force-to-far-field-pressure is given by Eq. (9.14). We calculate the transfer functions at 2049 evenly-sampled frequencies from 0 to 965.28 Hz and then window with the actuator/disturbance dynamics. At zero frequency (a degenerate point in the mathematical model) the transfer functions are taken to be zero.

The impulse responses are calculated by 4096-point (unscaled) FFT, and the results are downsampled by a factor of two. At the Nyquist frequency, 483 Hz, the transfer function amplitudes are down approximately 40 dB from their maxima. Using the actuator dynamics to roll off the transfer functions minimizes acausal leakage due to the frequency windowing; the time series are windowed to remove all remaining acausal artifacts. The resulting data has $l_e \approx l_s \approx l_d \approx 200$. We assume that this data is representative of the plant in question and that control of this discrete-time model indicates control of the actual system.

4.2. *Tradeoff Study*

The tradeoff study is accomplished by a sequence of controller syntheses, each with different constraint values. For the individual controller syntheses we solve

$$\begin{aligned} \min_{\mathbf{W}} \quad & \|\mathbf{E}_0 + \mathbf{D}_0 \circledast \mathbf{W} \circledast \mathbf{S}_0\|_2 \\ \text{subject to} \quad & \|\mathbf{E}_1 + \mathbf{D}_0 \circledast \mathbf{W} \circledast \mathbf{S}_1\|_2 \le \alpha_1, \\ & \|0 + \mathbf{D}_0 \circledast \mathbf{W} \circledast \mathbf{I}\|_2 \le \alpha_2, \\ & \|0 + \mathbf{I} \circledast \mathbf{W} \circledast \mathbf{S}_3\|_2 \le \alpha_3, \end{aligned} \tag{26}$$

where $\mathbf{E}_0$ is the disturbance-to-regulated-output response and $\mathbf{S}_0$ is the disturbance-to-control-sensor response both band-pass filtered emphasize the band of interest. $\mathbf{E}_1$ and $\mathbf{S}_1$ are band-stop filtered to emphasize out-of-band. $\mathbf{D}_0$ is the control-actuator-to-regulated-output, and $\mathbf{S}_3$ is the unfiltered disturbance-to-control-sensor. The minimization criterion is the in-band residual and represents the quieting power of the system. The first constraint limits the out-of-band radiation increase due to the control system. The second constraint limits the effect of sensor noise on the regulated output, and the third limits the RMS signal into the actuators for a given white noise input. This third constraint is useful for those types of actuators which may be able to handle large peak input signals, but can't handle a large persistent input as they will overheat.

Figure 4 shows the results of eight individual controller syntheses. For each synthesis we solve (26) for a different value of α_1. The x-axis gives out-of-band enhancement, $20\log_{10}(\alpha_1/\|\mathbf{E}_1\|)$, and the y-axis gives in-band residual, $20\log_{10}(\|\mathbf{E}_0 + \mathbf{D}_0\mathbf{W}\mathbf{S}_0\|/\|\mathbf{E}_0\|)$. Each asterisk (*) represents a different controller design, and the connecting lines interpolate the optimal tradeoff curve. Off the plot to the left, the out-of-band enhancement constraint is inactive, and the residual is limited by the sensitivity.

If plotted on these axes, any linear controller which satisfies the constraints will give a point on or above this optimal tradeoff curve. This curve shows that, for this plant, a significant reduction in out-of-band enhancement requires a smaller increase in residual.

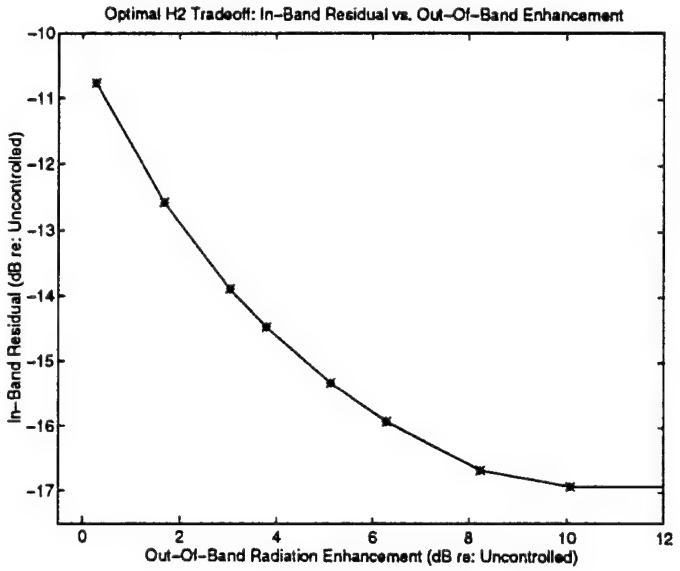

Figure 4. Optimal H_2 tradeoff study.

5. Conclusions

In this paper we solve a quadratically constrained quadratic problem, with multiple simultaneous constraints allowed. We treat this problem as a convex optimization problem on the Q-parametrization of the controller, particularly as a semi-definite program. Our algorithm of choice is to reduce a primal-dual potential function. Working on both the primal and dual problems together enables us to produce improved iterates for each problem, thus speeding up the convergence to the unique global minimizer. Additionally, a rigorous stopping criterion exists so that it is clear when the algorithm has converged. A preliminary first step to starting the algorithm either produces a feasible starting point or conclusively shows that the constraints are infeasible. Thus, for multiple H_2 constraints, either we will design the optimal discrete-time LTI controller to satisfy all the constraints or determine beyond doubt that no such realizable controller exists.

The computer-intensive portions of the algorithm are (i) solving a least squares problem for the primal direction, and (ii) computing two generalized eigenvalue problems in preparation for the plane search. We accelerate these two components, and greatly reduce the storage needed, by using Kronecker product identities, FFT's on extended matrices, and low-rank representations of structured matrices.

We present an example of examining the tradeoff between in-band residual and out-of-band enhancement while constraining two sensitivity measures. One constraint represents a physical limit on actuator authority, and the other constraint represents how we can limit the effects of sensor noise on the error microphone output. Studies like this are of obvious utility to control designers. Prior to the design formulation presented here, these studies were prohibitively expensive.

Acknowledgments

The authors wish to thank Stephen Boyd for his invaluable help with the theory and practical considerations regarding semi-definite programming.

Notes

1. It is important to note that $(\mathbf{E}_j, \mathbf{S}_j, \mathbf{D}_j)$ are place holders. Although we assign them a particular significance based on our controller design applications, they can be any impulse response matrices, even the zero or identity impulse response.
2. MATLAB is a registered trademark of The MathWorks, Inc., 24 Prime Park Way, Natick, MA 01760.

References

1. P.J. Titterton Jr., "Practical multi-constraint H^∞ controller synthesis from time-domain data," to appear in *International Journal of Robust and Nonlinear Control*, 1996.

2. P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, Academic Press, Harcourt Brace and Company, 1981.
3. S.P. Boyd, V. Balakrishnan, C.H. Barratt, N.M. Khraishi, X. Li, K.G. Meyer, and S.A. Norman, "A new cad method and associated architectures for linear controllers," *IEEE Transactions on Automatic Control.*, Vol. 33, No. 3, pp. 268–283, 1988.
4. T.F. Chan, J.A. Olkin, and D.W. Cooley, "Solving quadratically constrained least squares using black box solvers," *BIT*, Vol. 32, pp. 481–495, 1992.
5. J. Olkin, M. Freed, C. Gellrich, and P. Jungers, "SWAPS user's manual," *Internal Report of the System Technology Division*, SRI International, 1994.
6. C. Moler et al., *MATLAB User's Guide*, The MathWorks, Inc., Natick, MA, 1994.
7. S. Boyd and C. Barratt, *Linear Controller Design: Limits of Performance*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
8. L. Vandenberghe and S. Boyd, "Semi-definite programming," *Society for Industrial and Applied Mathematics Review*, Vol. 38, No. 1, pp. 49–96, 1996.
9. Y. Nesterov and A. Nemirovsky, *Interior Point Polynomial Algorithms in Convex Programming*, Society for Industrial and Applied Mathematics, 1994.
10. K.G. Oetzel, "A fast algorithm for calculating multi-channel wiener filters," *Internal Report of the System Technology Division*, SRI International, 1989.
11. M. Iri and H. Imai, "A multiplicative barrier function method for linear programming," *Algorithmica*, Vol. 1, pp. 455–482, 1986.
12. W.A. Weeks and B.L. Curless, "A real-time multichannel system with parallel digital signal processors," *Proceedings of the 1990 International Conference on Acoustics, Speech, and Signal Processing*, pp. 1787–1790, 1990.
13. M.C. Junger and D. Feit, *Sound, Structures, and Their Interaction*, The MIT Press, Cambridge, Mass., 1986.

Julia A. Olkin is a Research Engineer in the Applied Control and Signal Processing Group of SRI International in Menlo Park, CA. Her research interests include numerical linear algebra, convex optimization, controller design, and signal processing. During her eight years with SRI she has been heavily involved in algorithm development for controller synthesis in the area of active noise and vibration control.

Dr. Olkin received the B.A. degree (1981) from Pomona College in Mathematics and the M.S. (1986) and Ph.D. (1986) from Rice University, both in Mathematical Sciences.

Paul J. Titterton, Jr. is a Senior Research Physicist in the Applied Control and Signal Processing Group of SRI International in Menlo Park, CA. His research interests include network methods for sound-structure interactions and convex constrained optimization for active and passive control-law synthesis. In ten years with SRI, Titterton has been involved in a variety of projects, from modeling of atmospheric plasmas to algorithm development for multi-constraint control-law synthesis.

Dr. Titterton received an S.B. in Physics (1986) from MIT, an M.S. in Applied Physics (1989) from Stanford University, and a Ph.D. in Electrical Engineering (1996) also from Stanford University.

A Practical Method for Constrained-Optimization Controller Design: H^2 or H^∞ Optimization with Multiple H^2 and/or H^∞ Constraints

Paul J. Titterton, Jr.
pauljt@updike.sri.com

Julia A. Olkin
olkin@updike.sri.com

System Technology Division
SRI International
Menlo Park, CA, 94301

Abstract

This paper presents a practical algorithm for multi-input multi-output control law synthesis with an H^2- or H^∞-norm minimization criterion and multiple H^2- and/or H^∞-norm constraints. The plant is described by its discrete-time impulse response matrix, which can be determined directly from measurements. The control law design parameters are the tap weights of an FIR discretization of the Q-parameter. The multi-constraint H^2/H^∞ control law synthesis is written as a convex constrained optimization problem and solved as a structured semi-definite program. As a design example, we control the acoustic radiation from a (mathematically modeled) submerged spherical shell. The plant-model impulse response matrix has McMillan degree 800. We specify, synthesize, and compare three controllers: one designed by standard H^2 techniques; one designed by H^∞ minimization with three H^∞ and three H^2 constraints; and one designed by H^2 minimization with the same constraints. In a single design iteration, the multi-constraint H^2/H^∞ controllers achieve the best possible performance given the constraints; after several design iterations, the standard H^2 controller achieves slightly worse performance with several constraint violations.

1 Introduction

This paper describes a practical method for discrete-time linear feedback control law synthesis with an H^2- or H^∞-norm minimization criterion and multiple H^2- and/or H^∞-norm constraints. Many high-level specifications on the controlled system—stability robustness and actuator authority limits, for example—can be written as simultaneous constraints on the H^2 and H^∞ norms of the controlled-plant transfer matrices [1]. The method described in this paper enables direct translation of those specifications into a control law. This translation is demonstrated in the design example: multi-input multi-output (MIMO) reduction of acoustic radiation from a submerged spherical shell. We impose simultaneous constraints on the stability robustness measure, performance sensitivity, actuator authority, and out-of-band radiation enhancement. This example also demonstrates our method's ability to synthesize control laws for realistically large, dynamically complex plants.

We assume that the plant is well represented by a discrete-time finite-length impulse response (FIR) matrix. The control law synthesis uses the FIR model directly; that is, the control law synthesis requires only minimally processed measured data. There is no need for analytic model generation, state-space system identification, or model reduction.. The FIR model does, however, restrict direct application of this synthesis method to stable plants with relatively short impulse responses. We note that with a preliminary stablizing controller design [2], this restriction can be lifted.

For control law synthesis, we use the simplest possible Q-parameterization for stable plants. As per Boyd and Barrett [1] (Section 7.3.1), a parallel and negated transmission path in the control hardware neutralizes the control-actuator-to-control-sensor transfer function. We follow Boyd et al. [2] and Peterson, Weeks, and Nowlin [3] in using a matrix FIR representation for the Q-parameter, with the FIR's individual tap weights as the design variables.

Several previous design algorithms exist for the FIR plant model and Q-parameterization. Boyd et al. [2] developed a general control design compiler, `qdes`, around a linearly constrained quadratic optimization engine. Olkin et al. [4] developed a (single constraint) quadratically constrained quadratic optimization engine that exploits the problem structure. The latter code was used by Peterson [3] and is used for this paper's preliminary H^2 design. Titterton [5] developed a multi-constraint H^∞ design algorithm, and Olkin and Titterton [6] developed an algorithm for multi-constraint H^2 with H^2 minimization.

The current controller design methods of choice—the coupled Ricatti equation methods [7]—find state-space controllers which minimize the H^2 or H^∞ norm of a single closed-loop transfer function (CLTF) with *no* constraints. That is, specifications on the stability robustness and actuator authority, for example, cannot be individually enforced. To achieve specifica-

1058-6393/96 $5.00 © 1996 IEEE
Proceedings of ASILOMAR-29

tions, the control designer heuristically and iteratively modifies pre- and post-weighting matrices, which emphasize (or de-emphasize) parts of the CLTF. If the specifications are not met after some number of weight matrix iterations, the designer will not know if the specifications are actually infeasible or merely difficult to achieve.[1]

In a recently published paper, Scherer [8] provides a theoretical convergence analysis of a multi-constraint H^2/H^∞ method similar to that of this paper. However, no practical algorithm for solving realistically sized problems is given, and he uses MATLAB's Linear Matrix Inequality Toolbox [9] to calculate the simple numerical example. For small problems, the MATLAB toolbox can also implement this paper's algorithm, but the data matrices grow quickly with problem size, and special purpose code is required (depending on computational resources) well before the control law for our acoustic radiation problem can be synthesized.

This paper's contribution is the practical method for large-scale multi-constraint H^2/H^∞ control law synthesis. The main advantage of this method, from a control designer's point of view, is that multiple constraints simplify the translation from controlled-plant specifications to a control law synthesis problem. The design approach is more direct, and control engineers can solve larger and more complicated problems.

The work presented here constitutes a major part of our larger effort to develop a practical "specification-based" controller synthesis method [4, 6, 5]. We plan to add ℓ_∞- and ℓ_1-norm minimization and constraints. These additional norms will enable time-domain specifications such as: Given a time-domain peak acoustic disturbance of 10μPa, limit the peak actuator input to 5V.

The outline of this paper is roughly parallel to that of a typical design procedure: (1) write the controlled-plant specifications as an H^2/H^∞ constrained optimization (CO) problem; (2) transform the CO problem into a *convex* CO problem by Q-parameterization of the controller; and (3), translate the convex CO into a semi-definite program (SDP) and solve for the Q-parameter. In Section 2, we briefly describe the mathematical model which generates the plant data, define the H^2 and H^∞ norms, and state high-level specifications on the controlled plant as a CO problem in terms of those norms. In Section 3, the Q-parameterization transforms the CO problem into a convex CO problem, and in Section 4, we compare the multi-constraint H^2/H^∞ design to a standard H^2 design.

2 Plant model and specifications

As a design example, we will reduce the in-band acoustic radiation from a submerged spherical shell while simultaneously satisfying six constraints: (1) stability robustness, (2 and 3) performance sensitivity to transfer function perturbations, (4 and 5) individual actuator authority, and (6) out-of-band enhancement.

With reference to Figure 1, the radiating sphere (the plant) has two disturbance sources, two control sensors, two control actuators, and two radiated field (regulated output) sensors. We assume that the plant dynamics are fully described by four finite-length impulse response matrices: **E** is the disturbance-to-regulated-output response, **S** is the disturbance-to-control-sensor response, **D** is the control-actuator-to-regulated-output response, and **C** is the control-actuator-to-control-sensor response. (Bold capital letters represent FIR matrices.) The impulse-response matrices are all about 200 samples long at a sample rate of about 1 kHz. We assume that this data is representative of the plant in question and that control of this discrete-time model indicates control of the actual system.[2]

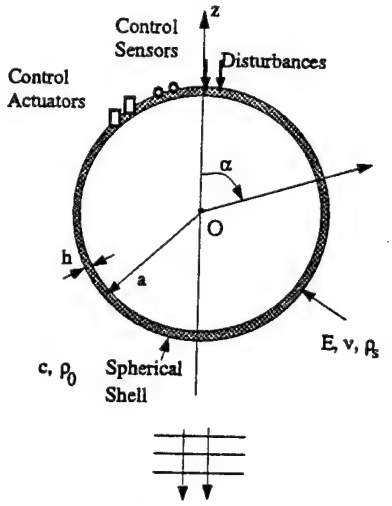

Figure 1: Submerged Spherical Shell: The plant has two disturbances, two control sensors, two control actuators, and two radiated-field sensors. We will reduce the disturbances' acoustic radiation while simultaneously constraining a stability robustness measure, performance sensitivity, actuator authority, and out-of-band enhancement.

High-level specifications on the controlled system are as follows:

- Reduce acoustic radiation, as much as possible, over the band 40 to 180 Hz.
- Guarantee closed-loop stability for a specified level of output-multiplicative uncertainty in **C**.
- Constrain (to α_2) the closed-loop transfer function's sensitivity to additive uncertainty in **D**.

[1] In the multi-constraint H^2/H^∞ method, weighting matrices can be used to shape the frequency response and emphasize input and output combinations for any of the constrained CLTFs. Because the weightings are applied constraint by constraint, their choice is simple and direct.

[2] Reference [6] describes the shell model and processing which generate the plant data.

- Constrain (to α_3) the closed-loop transfer function's sensitivity to additive uncertainty in $\mathbf{S}$.
- Limit (to α_4) the RMS response of actuator 1 to unit-variance white noise disturbance inputs.
- Limit (to α_5) the RMS response of actuator 2 to unit-variance white noise disturbance inputs.
- Limit (to α_6) the white-noise-to-RMS out-of-band response.

In general, as the constraints are tightened or relaxed, the optimal value of the cost function rises or falls. This enables us to calculate optimal tradeoff curves which explore the envelope of control system performance.

The above list describes typical noise/vibration control specifications. These specifications can be quantified using the H^2 and H^∞ norms, which are described in references [6] and [5]. The H^2 norm of $\mathbf{H}$ is the RMS output given unit-variance white noise input. The H^∞ norm of $\mathbf{H}$ is the worst-case RMS output given any unit-RMS input.

2.1 Synthesis problem formulation

Given these H^2- and H^∞-norm definitions, we write down a CO problem which quantifies the controlled-plant specifications given above:

$$\begin{array}{ll} \text{minimize} & \|\mathbf{H}_0\|_{2 \text{ or } \infty} \\ \text{subject to} & \|\mathbf{H}_j\|_\infty \le \alpha_j, \text{ for } j = 1,2,3 \\ & \|\mathbf{H}_j\|_2 \le \alpha_j, \text{ for } j = 4,5,6, \end{array} \qquad (1)$$

where

- $\mathbf{H}_0$ is the closed-loop disturbance-to-regulated-output FIR matrix, including the filter which emphasizes the in-band response.
- $\mathbf{H}_1$ is the closed-loop actuator-input-to-controller-output FIR matrix. We set $\alpha_1 = 5$, which guarantees stability for any multiplicative uncertainty with H^∞ norm less than 0.2.
- $\mathbf{H}_2$ is the closed-loop disturbance-to-control-output FIR matrix.
- $\mathbf{H}_3$ is the closed-loop control-input-to-regulated-output.
- $\mathbf{H}_4$ is the closed-loop disturbance-to-actuator-1.
- $\mathbf{H}_5$ is the closed-loop disturbance-to-actuator-2.
- $\mathbf{H}_6$ is the closed-loop disturbance-to-regulated-output, including the filter which emphasizes the out-of-band response. We choose $\alpha_6 = 7500$, which allows the controller-on $\|\mathbf{H}_6\|_2$ to be twice the controller-off $\|\mathbf{H}_6\|_2$.

Reasonable sensitivities and actuator authorities are given by $\alpha_2 = 575$, $\alpha_3 = 0.237$, and $\alpha_4 = \alpha_5 = 1725$. These values depend on transfer function units and are given for completeness only. The filter which emphasizes the in-band response in $\mathbf{H}_0$ is a fifth-order band-pass type-1 Chebychev with 0.5 dB pass-band ripple and cutoff frequencies 40 and 180 Hz. The filter which emphasizes the out-of-band response in $\mathbf{H}_6$ is a fifth-order band-stop type-1 Chebychev with 0.5 dB pass-band ripple and cutoff frequencies 50 and 150 Hz.

3 Controller parameterization

A block diagram of the plant and controller is shown in Figure 2. The controller is written in terms of two FIR filters: $\mathbf{N}$ and $\mathbf{Q}$. We set the first to the opposite of the nominal control-actuator-to-control-sensor response, $\mathbf{N} = -\mathbf{C}$. This is the simplest possible Q-parameterization for stable plants. The control design parameters are the tap weights of $\mathbf{Q}$, the FIR representation of the Q-parameter.

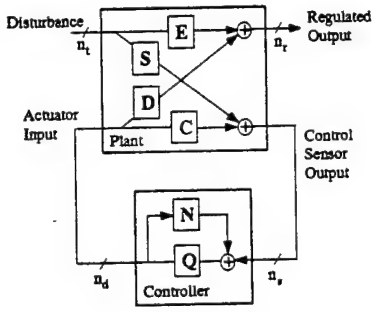

Figure 2: Plant and Controller Topology and Notation

The Q-parameterization has two attributes which make it a good choice for the design parameter. First, as $\mathbf{Q}$ ranges over all possible stable response functions, the controlled plant ranges over all possible stabilizable closed-loop transfer functions. Second, closed-loop impulse responses are affine in $\mathbf{Q}$; for example, the disturbance-to-regulated-output response is

$$\mathbf{H} = \mathbf{E} + \mathbf{D} \circledast \mathbf{Q} \circledast \mathbf{S}, \qquad (2)$$

where $\circledast$ means discrete-time linear convolution, and the closed-loop responses in Eq. (1) can be written in this form. We should note that Q's FIR discretization is limiting: The length of $\mathbf{Q}$ required to achieve the optimum performance for any linear controller is not clear *a priori*. Experience indicates that a good first choice is the length of the plant data. If, after solving Eq. (1), the Q-parameter time series do not decay to zero, we can gain design freedom by lengthening $\mathbf{Q}$.

We apply H^2 and H^∞ norms to closed-loop quantities that are affine in $\mathbf{Q}$. Thus Eq. (1) is a *convex* CO problem. It is this convexity that makes the optimization numerically (and analytically) tractable.

The convex CO is translated into an SDP, which can be solved using recently developed interior-point methods [10]. This translation is briefly described in References [6] and [5].

4 MIMO control of radiated sound

This section first solves Eq. (1) with the H^∞-norm minimization criterion, and then with the H^2 minimization criterion. Results are compared to those for a standard H^2 design (discussed in Appendix A).

In each case we choose the Q-parameter to be a 130-tap FIR. Because the calculated Qs all decay to zero, we are assured that no additional design freedom can be gained by lengthening the FIR.

The solution is calculated via an interior-point method, and as such, requires a convergence criterion for exit. We set the convergence criterion so that any additional optimization could improve the results by at most one percent.

The solution method also requires an initial feasible point. In Eq. (1), $\mathbf{Q} = 0$ is sufficient. If the initial $\mathbf{Q}$ were non-trivial, a series of Phase 1 CO problems [11] would determine the starting point. If this process determined a null feasible set, then we would have identified our specifications as inconsistent.

4.1 Multi-const. with H^∞ minimization

Figure 3 shows the results of multi-constraint design with an H^∞ minimization criterion. The plot shows the maximum singular values of the unfiltered disturbance-to-regulated-output transfer function for: (1) controller off, i.e. the open-loop response, (2) controller on using standard H^2 design for $\mathbf{Q}$, and (3) controller on using multi-constraint with H^∞ for $\mathbf{Q}$. The flat trace over the performance band is typical of H^∞ control law designs: We minimize the maximum singular value over frequency. The H^∞-norm radiation reduction (that is, the Eq. (1) cost function reduction) is 22 dB. We should note that, at the edges of the band of interest, the standard H^2 design reduces the maximum singular value more than the H^∞. This effect is expected: To minimize the maximum singular value in the band's center, the band edges are increased. The H^∞-norm radiation reduction for the standard H^2 design is 14 dB.

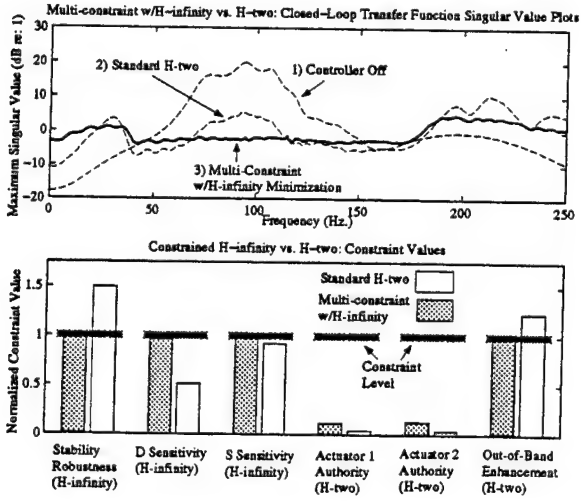

Figure 3: Multi-constraint H^2/H^∞, with H^∞ minimization, compared to standard H^2 controller synthesis. The singular value plot shows disturbance-to-regulated-output transfer functions, and the bar chart shows the normalized values of the constrained transfer functions.

The bar chart shows the (normalized) values of the constrained transfer functions. The dark horizontal lines are the constraint values, normalized to one; the shaded bars are the norms of the constrained transfer functions for the multi-constraint with H^∞; and the unshaded bars are the norms of the same transfer functions for the standard H^2. The standard H^2 violates the stability robustness and out-of-band enhancement constraints, while giving up design freedom in the sensitivity and actuator authority constraints. The multi-constraint uses all the design freedom in the stability robustness, sensitivity, and out-of-band enhancement specifications, while the actuator authority constraints are inactive.

4.2 Multi-const. with H^2 minimization

Figure 4 shows the results of multi-constraint design with the H^2 minimization criterion. The plot shows the maximum singular values of the unfiltered disturbance-to-regulated-output transfer function for: (1) controller off, (2) controller on using standard H^2 design for $\mathbf{Q}$, and (3) controller on using multi-constraint with H^2-minimization for $\mathbf{Q}$. The H^2-norm radiation reduction for the multi-constraint H^2 is 16 dB. For the standard H^2, it is 13 dB.

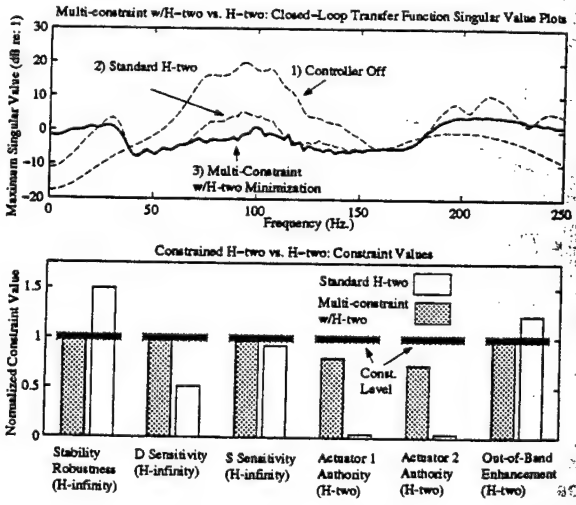

Figure 4: Multi-constraint H^2/H^∞, with H^2-minimization, compared to standard H^2 controller synthesis. The singular value plot shows disturbance-to-regulated-output transfer functions, and the bar chart shows the normalized values of the constrained transfer functions.

The bar chart shows the (normalized) values of the constrained transfer functions. The shaded bars are the norms of the constrained transfer functions for the multi-constraint with H^2, and the unshaded bars are the norms of the same transfer functions for the standard H^2. The multi-constraint uses all the design freedom in the stability robustness, sensitivity, and out-of-band enhancement specifications. The actuator authority, while the constraints are still inactive, is significantly larger than the standard H^2 results.

This increase is the price of the extra 3 dB radiation reduction.

5 Conclusion

The practical algorithm presented in this paper enables us to specify and simultaneously satisfy multiple H^2 and H^∞-norm constraints while minimizing an H^2 or H^∞-norm cost function. FIR matrices describe the plant and the Q-parameter. This is a natural description for many systems and can be taken almost directly from measured data. The tap weights of Q are the controller design parameters.

The Q-parameterization and convex optimization guarantee that our control laws yields the best possible performance given the constraints. This is in contrast to the standard H^2 design, in which we adjusted the scale factors by hand until the performance was reasonable and only a few constraints were violated. When compared to any linear controller that meets the constraints, designed by any other means, the multi-constraint H^2/H^∞ control law will have equal or better performance.

Our method synthesizes a control law which directly satisfies controlled-plant specifications. This method can also examine optimal tradeoffs between specifications, which is of obvious utility to control designers.

We will continue our efforts in specification-based control law synthesis by adding ℓ^1- and ℓ^∞-norm minimization criteria and constraints. These norms will enable time-domain signal amplitude minimization and constraints.

A Standard H^2 design

The H^2 design is accomplished using code developed by Olkin et al. [4], which yields the $\mathbf{Q}^*$ that minimizes

$$\begin{array}{rcl} & & \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=0}^{l_0}\{\mathbf{H}_0^{(i,j)}(k)\}^2 \\ + & 0.1\mu & \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=0}^{l_q-1}\{\mathbf{Q}^{(i,j)}(k)\}^2 \\ + & 0.1 & \sum_{i=1}^{n_z}\sum_{j=1}^{n_w}\sum_{k=0}^{l_1}\{\mathbf{H}_1^{(i,j)}(k)\}^2 \end{array} \qquad (3)$$

where k runs from zero to the closed-loop impulse response lengths, and

$$\mu = \max_{i'j'k'} \frac{\partial^2}{\partial(\mathbf{Q}^{(i',j')}(k'))^2} \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=0}^{l_0}\{\mathbf{H}_0^{(i,j)}(k)\}^2.$$

The second term in (3) regularizes the solution by weighting the 2-norm of Q by 0.1μ. The form of μ scales the weighting to the problem at hand, and the assignment of the scale factor (0.1) is entirely heuristic: Experience indicates that this level of regularization gives fair performance with fair robustness. The third term weights the closed-loop matrix which is constrained for stability robustness: We expect this term to add robustness with out much effect on performance.

We iteratively adjusted the weighting coefficients until the control system performance shown in Figures 3 and 4 were achieved. The final coefficient values, 0.1μ and 0.1, were achieved after about 10 iterations. As is characteristic with standard design methods, we do not know whether additional weighting or additional adjustments would meet the constraints *and* provide sufficient radiation reduction.

Acknowledgments

The authors are grateful to Stephen Boyd for invaluable help with the theory and practical considerations regarding semi-definite programming. This work was supported by the Office of Naval Research under Contract N00014-94-C-0128.

References

[1] Stephen Boyd and Craig Barratt. *Linear Controller Design: Limits of Performance.* Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[2] Stephen. P. Boyd, Venkataramanan Balakrishnan, Craig H. Barratt, Nasser M. Khraishi, Xiaoming Li, David G. Meyer, and Stephen A. Norman. A new CAD method and associated architectures for linear controllers. *IEEE Transactions on Automatic Control*, 33(3):268–283, 1988.

[3] D. Kent Peterson, William A. Weeks, and William C. Nowlin. Active control of complex noise problems using a broadband, multichannel controller. In *Noise-Con 94*, pages 315–320, 1994.

[4] J. A. Olkin, M. Freed, C. Gellrich, and P. Jungers. *SWAPS User's Manual.* System Technology Division, SRI International, 1994.

[5] Paul J. Titterton, Jr. Practical multi-constraint H^∞ controller synthesis from time-domain data. *The International Journal of Robust and Nonlinear Control.* To Appear in 1996.

[6] Julia Olkin and Paul J. Titterton, Jr. Using semi-definite programming for H^2 controller design with multiple simultaneous H^2 constraints. *The Journal of VLSI Signal Processing.* Submitted 1995.

[7] John C. Doyle, Keith Glover, Pramod P. Khargonekar, and Bruce A. Francis. State-space solutions to standard H^2 and H^∞ control problems. *IEEE Transactions on Automatic Control*, 34(8), August 1989.

[8] C. Scherer. Multiobjective H^2/H^∞ control. *IEEE Transactions on Automatic Control*, 40:1054–1062, June 1995.

[9] P. Gahinet, A. Nemirovski, A. Laub, and M. Chilali. *MATLAB LMI Control Toolbox.* The Mathworks, Inc., 1995.

[10] Y. Nesterov and A. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming: Theory and Application.* Springer-Verlag, 1993.

[11] L. Vandenberghe and S. Boyd. Semi-definite programming. *To appear in SIAM Review*, 1995.

Technical Note • December 1995

SELECTION OF NEUTRALIZATION FOR CONTROL-SYSTEM STABILITY ROBUSTNESS

Prepared by:

RICHARD P. HEYDT, *Research Engineer*
Applied Control and Signal Processing Group
System Technology Division

Prepared for:

Office of Naval Research
Code 1222
800 N. Quincy St.
Arlington, VA 22217

Attn: Dr. Geoffrey L. Main
Program Manager, Structures

Contract N00014-94-C-0128

SRI Project 5796

A technical note is a working paper that presents the results of research related to a single phase or facet of a research problem. It presents the concepts, findings, and/or conclusions of the authors, alone, in fragmentary form intended for internal discussion and comment. Such notes are subject to revision, incorporation into more formal reports, or withdrawal.

Approved by:

J. RAUL MARTINEZ, *Director*
Applied Control and Signal Processing Group

SUMMARY

This report documents recent results on the subject of control system performance in the presence of variability. Coupling variability must be accounted for in controller design if stability is to be maintained, and this exacts a price on system performance. The coupling neutralization is chosen as part of the design process, and an important issue is to determine what effect, if any, the choice of neutralization has on performance. This issue is investigated by designing example controllers based on the CT-B data set (24 hours' worth of coupling data) and taking into account the observed measurement variability. The effect on performance is illustrated for three choices of coupling neutralization, and the results show that proper selection of neutralization can mean the difference between marginal performance (3 dB) and good performance (> 8 dB). The work here expands on preliminary work by Paul Titterton and William Nowlin[1] on the tradeoff of control-system performance and stability robustness, and is part of a continuing investigation into the design of optimal Q-parameter controllers with real data.

The neutralization selection problem we discuss is properly formulated as a convex program. As such, it was readily solved using SRI's mixed norm constrained optimization program, MINCODE, showing that MINCODE's applicability is not limited to control law design.

[1] Titterton, P.J. and Nowlin, W.C., "Optimal Performance/Robustness Tradeoff Using CT-B Data With Preliminary Control Approach," Applied Control and Signal Processing Group client briefing, 18 April 1995.

1 INTRODUCTION

Figure 1 is a block diagram of the neutralized feedforward, or Q-parameter, control system architecture. The plant transfer functions, **E**, **S**, **D**, and **C**, and the controller transfer functions, **W** and **N**, are represented by FIR filters. Ideally, the transfer function **N** is set equal to **–C** to obtain "perfect" neutralization. The controller design problem is to select a weight set, **W**, that minimizes the effect of an external disturbance on the system. Under the assumption of perfect neutralization, the design problem is thus

$$\min_{\mathbf{W}} \|\mathbf{E} + \mathbf{DWS}\|_{\infty} \tag{1}$$

Note that we have selected the function to be minimized as an infinity-norm, i.e., we are minimizing the maximum RMS gain (over all frequencies) of the closed-loop transfer function. We could choose other norms, such as the 2-norm, as the objective function. This would not change the analysis approach, nor would it change the conclusions suggested by the analysis.

Implicit in the Q-parameter controller design problem is that the plant transfer functions can be, and generally are, derived directly from measured data. This means that **E**, **S**, **D**, and **C** all have inherent variability and that for a given disturbance and selected **W** the actual performance will vary over some range. The focus here is on the effect of variability in the coupling transfer function, **C**.

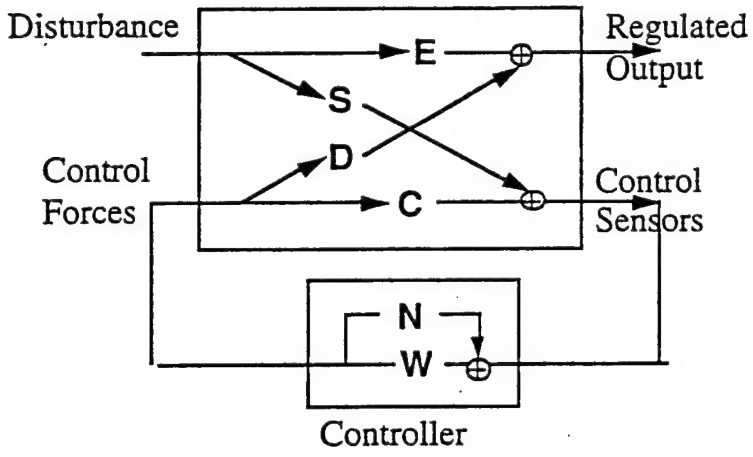

Figure 1. Q-Parameterized Control System

The coupling variability is represented as an additive factor, $\delta\mathbf{C}$, as indicated by Figure 2. If the nominal coupling transfer function is **C**, the real coupling at any moment in time, or for any particular measurement set j, is $\mathbf{C} + \delta\mathbf{C}_j$. Note that in the analysis of Reference [1], the variability was assumed to be multiplicative, so that the jth coupling transfer function was $\mathbf{C}\delta\mathbf{C}_j$. The additive variability representation is used for convenience in this study, because it simplifies determination of the factor $\delta\mathbf{C}$. However, there is no loss of generality.

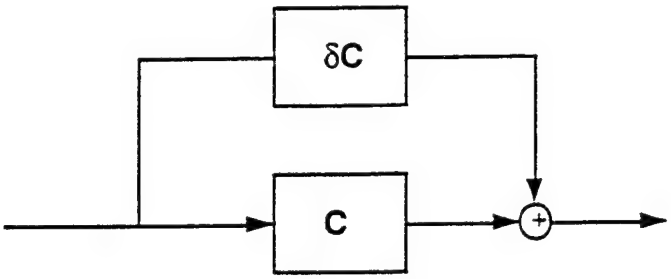

Figure 2. Model for Additive Coupling Variability

Variability in any of the plant transfer functions affects control-system performance, but coupling variability affects both performance and stability. For additive variability, it follows from the small-gain theorem that, as a sufficient condition for the control system to be stable, the weight set W must satisfy

$$\forall j \quad \| W\delta C_j \|_\infty < 1 \tag{2}$$

The inequality must hold for all possible values of the variability δC_j. If a series of plant measurements is made to characterize the coupling variability, then the subscript j can be considered as the measurement-set index. In this case, the stability requirement can be enforced (conservatively) as

$$\|W\|_\infty \cdot \max_j \| N - C_j \|_\infty < 1 \tag{3}$$

or

$$\|W\|_\infty < (\max_j \| N - C_j \|_\infty)^{-1} \tag{4}$$

where N represents the nominal coupling transfer function. From a practical standpoint, it is essential that the coupling measurements fully span the variability space if stability is to be guaranteed during controller operation.

Since Equation (4) is a restriction on the infinity-norm of the weights, it will limit control-system performance. The impact on performance can be mitigated if the quantity in parentheses is made small. The question that we address here by example is, given a set of C measurements that characterize the actuator-to-sensor coupling, how important is the selection of the nominal coupling, N, to system performance?

2 CONTROL-DESIGN EXAMPLE PROBLEM

The example problem used here to address the issue of nominal coupling selection is the same problem used by Paul Titterton in [1]. This is a 2-input, 2-output control system developed from the CT-B data set, in which two control sensors are selected from a set of 16, and two grouped actuators are selected from a set of 11. In addition, two equivalent performance sensors are formulated from the total sensor set. The controller is intended to minimize the responses at the performance sensors when the system is subjected to external excitation. For the CT-B model this corresponds to minimizing particular types of motion on the body.

The control problem is formulated as an objective function with two constraints:

$$\min_{W} \| (E + DWS) f_{PB} \|_{\infty} \tag{5}$$

subject to

$$\|W\|_{\infty} < (\max_{j} \| N - C_j \|_{\infty})^{-1} \qquad j = 1, 2, .. , \# \text{ data sets} \tag{6}$$

and

$$\|W f_{OB}\|_{\infty} < 3 \tag{7}$$

The performance band for the problem is 1.0 to 3.5 kHz, so the minimization in Equation (5) is restricted to this band with the filter f_{PB}. The first constraint ensures stability in the presence of additive variability, and the index runs from 1 to the number of coupling-data measurements. In the second constraint, f_{OB} represents an out-of-band filter. The second constraint restricts out-of-band enhancement to an arbitrary maximum.

3 CT-B COUPLING DATA

There are 24 sets of plant measurements in the CT-B data set, including actuator-to-sensor coupling. The measurement sets were made at 24 consecutive 1-hour intervals. Each coupling measurement consists of the time responses at 16 sensors due to excitation by each of 11 actuators. The full deconvolved coupling is therefore 24 measurements of an 11 x 16 matrix of impulse responses. For this analysis the responses were filtered with a 100-Hz high-pass filter to eliminate frequencies at which the actuator excitation and/or coupling signal-to-noise ratio were low.

The coupling of interest for the controller design problem is a 2-actuator-by-2-sensor subset of the full 11 x 16 coupling matrix. Figure 3 plots the infinity-norm of the quantity $\mathbf{N} - \mathbf{C}_j$ for each of the 24 measurements of this 2 x 2 coupling matrix. $\mathbf{N}$ is the nominal coupling, which was taken arbitrarily to be the 13th measurement set (the measurement made approximately halfway through the 24-hour period).

An interpretation of the infinity-norm is the maximum, over all frequencies, of the maximum singular value at each frequency. The infinity-norm of $\mathbf{N} - \mathbf{C}$ is therefore a measure of the maximum deviation over all frequencies of the coupling transfer function, $\mathbf{C}$, from the nominal coupling, $\mathbf{N}$. The quantity

$$\max_{j} \| N - C_j \|_{\infty} \tag{8}$$

can be thought of as the radius of a sphere centered at N, within which the measured variability for every data set j lies.

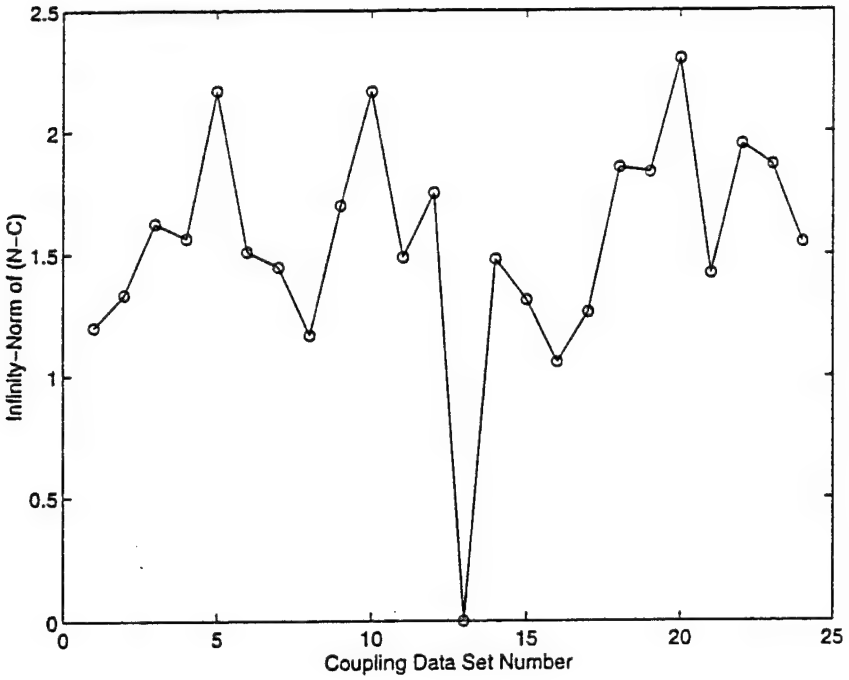

Figure 3. Infinity-norm of the quantity $N - C_j$, where C_j is the jth measurement of the 2 x 2 coupling transfer function from the CT-B data, and N is the nominal coupling, which here is the 13th coupling measurement.

We see in Figure 3 that the variability "radius" for the 24 coupling measurements ranges from about 1.2 to 2.3. The changes in radius from one data set to the next appear to be more or less random. The mean radius is 1.54, and there is no definitive increase in the mean value with time (data set number), which would indicate a change in the plant. This type of plot is very useful for identifying plant changes, as well as for identifying anomalous data sets or outliers.

Since the infinity-norm of the weights, $\|W\|_\infty$, in the example control problem is constrained to be less than the inverse of the coupling variability, the constraint level based on the data in Figure 3 would be the inverse of the maximum variability radius, or approximately (1/2.3).

As an aside, the assumption here is that the variability measurements are representative of the actual plant. This requires that the measurements be made sufficiently often to capture the full range of variation in coupling, and over a sufficient duration to capture any long-term drift. These are not trivial requirements. Determining the appropriate coupling-variability radius requires understanding the measured data from a statistical point of view and accounting for factors such as the time interval between system recalibrations. If there are anomalous data points in plots like Figure 3, they may be discounted in estimating the maximum variability radius. Alternatively, the radius may be assigned a probabilistic confidence level. These issues are in the realm of data analysis and system identification, but are clearly important for controller design.

The hypothesis is that it is advantageous to minimize the maximum variability radius in designing the controller. The nominal plant in Figure 3 was one of the actual coupling measurements. An obvious better choice is the average coupling, i.e., the coupling whose impulse response is the mean of the 24 measurements at each point in time (each tap). Figure 4 shows the infinity-norm of N – C when N is equal to the average coupling. In

this plot the mean variability radius is about 1.2 and the maximum radius is 1.7. While this is better than before, our choice of N is still ad hoc.

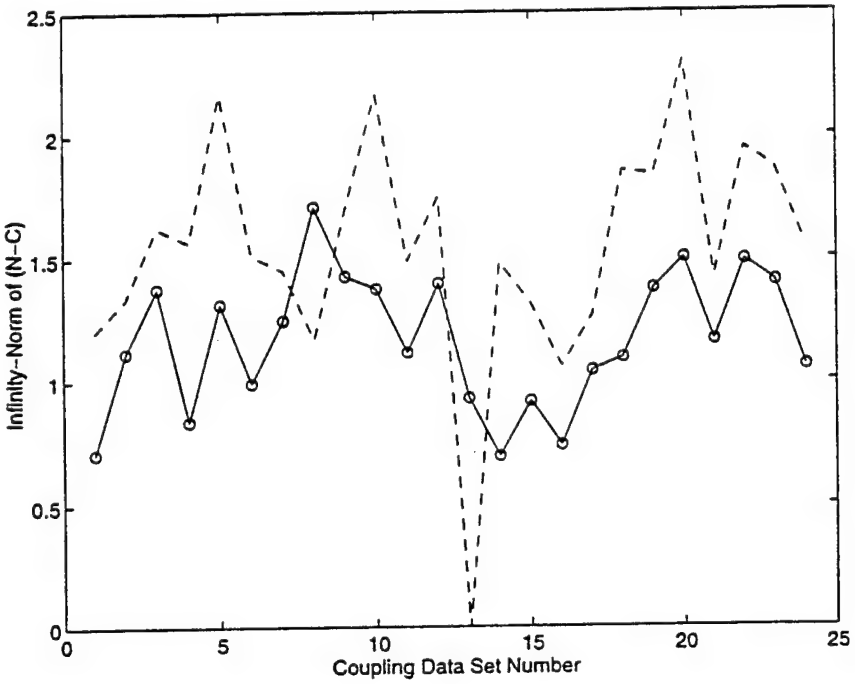

Figure 4. Infinity-norm of the quantity $\mathbf{N} - \mathbf{C}_j$, when **N** is the average coupling (solid line) and the 13th coupling measurement (dashed line).

The "proper" way to address the problem is to define an optimality criterion for choosing N. One such criterion would be to minimize the maximum variability radius. Finding the transfer function, N, that minimizes the maximum variability radius is a convex optimization problem, so we used a version of MINCODE to determine this N for the CT-B data set. (Note that the general convex optimization structure of MINCODE allows it to be applied to this important class of problems, which is not control-law design, per se). Figure 5 plots the infinity-norm of $\mathbf{N} - \mathbf{C}$ for the optimal N. For the optimal neutralization, the mean radius is reduced to 0.58 and the maximum radius to 0.74.

4 CONTROL-SYSTEM PERFORMANCE

Figure 6 shows a plot of control-system performance for the example problem. The performance, defined as

$$20 \log_{10} \left(\frac{\| \mathbf{E} + \mathbf{DWS} \|_\infty}{\| \mathbf{E} \|_\infty} \right) \tag{9}$$

is plotted as a function of the infinity-norm of W. This plot represents the nominal performance assuming perfect neutralization, and is directly analogous to the plot of performance vs. stability robustness developed by Titterton in [1]. The difference is that the Titterton plot is based on multiplicative variability, while Figure 6 is based on additive variability. As with the Titterton plot, performance improves as one moves down the y-axis, while robustness to variations in the coupling increases as one moves to the left on the x-axis. The latter follows directly from the stability design constraint,

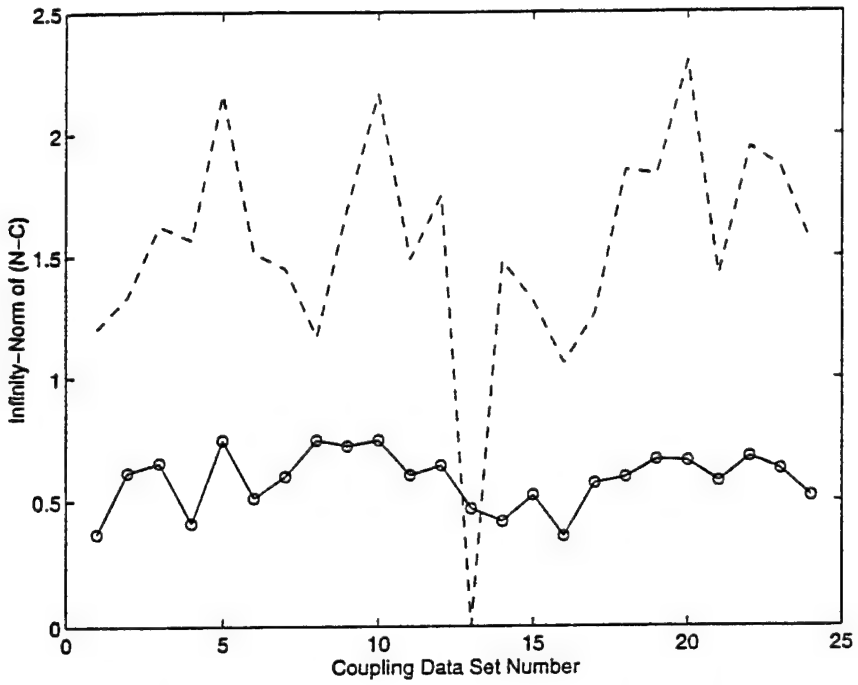

Figure 5. Infinity-norm of the quantity $\mathbf{N} - \mathbf{C}_j$, when **N** is the optimal coupling (solid line) and the 13th coupling measurement (dashed line).

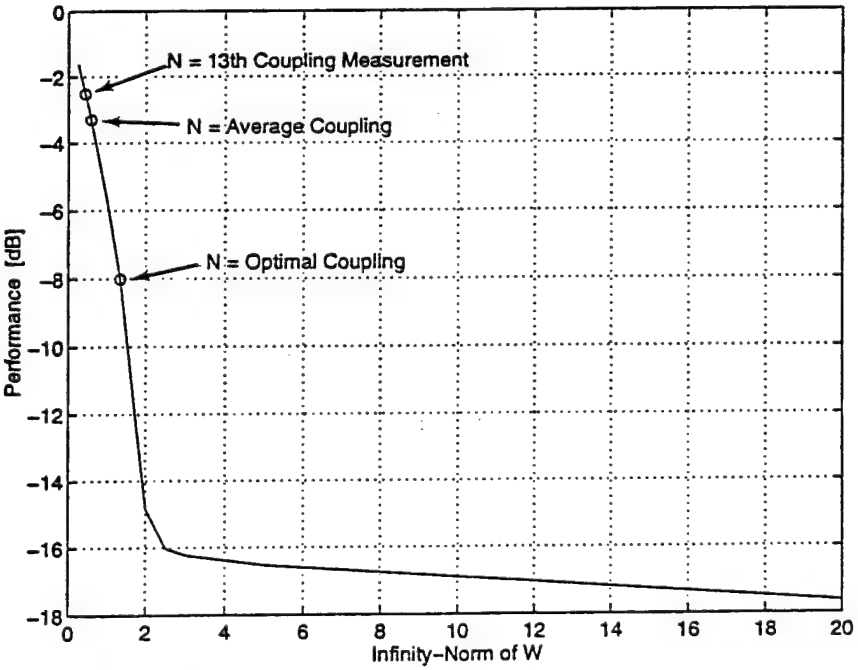

Figure 6. Control-system performance as a function of the stability constraint level on the coupling (Equation [6]). Performance increases moving down the vertical axis, and stability robustness increases moving left on the horizontal axis.

Equation (6). Note that the meaning of the x-axis value is different than in the case of multiplicative variability. For example, an x-axis value of 2 in Figure 6 does not mean that the system will be stable for 50% variations in the infinity-norm of the coupling.

The solid line in Figure 6 is the locus of performance for MINCODE-generated solutions to the controller-design problem. It is important to understand that this curve is completely independent of both **N** and **C**. It represents the optimum performance for a given value of the right-hand-side of Equation (6), the stability constraint. The graph shows that performance does indeed improve as the stability constraint eases, i.e., as the norm of **W** is allowed to become large. The incremental performance increase is marginal when the stability constraint is greater than about 2.5, but performance is very sensitive to the constraint at lower values.

In our control designs, we always constrain the solution appropriate to the observed level of variability—in this case as indicated by the maximum variability radius. Hence we can compare the performance of appropriately constrained controllers for each of the neutralizing transfer matrices (**N**) we have chosen. The three discrete points on the curve in Figure 6 are based on the maximum variability radii from Figures 3, 4, and 5. For example, if we take the 13th coupling measurement as the neutralization, the stability constraint that encompasses all of the measured data is

$$\|\mathbf{W}\|_\infty < \left(\max_j \|\mathbf{N} - \mathbf{C}_j\|_\infty\right)^{-1} = (2.3)^{-1} = 0.435 \qquad (10)$$

The performance in this case is only –2.5 dB, which reflects the fact that there is a considerable range over which the coupling is permitted to vary, and that no effort was expended to determine a good **N**. With the average coupling used as **N**, the performance improves to only –3.3 dB. However, if we choose the neutralization via optimization (MINCODE), **there is a substantial performance improvement to –8.0 dB**. In summary, we have designed three control laws that are stable in the presence of all measured variability. Depending on how the variability is apportioned between **N** and $\{\delta C_j\}$, we obtain between –2.5 and –8.0 dB of performance for otherwise identical designs.

The data of Figure 6 are replotted in Figure 7, with the x-axis changed to the maximum variability radius (Equation [8]), i.e., the inverse of the x-axis values in Figure 6. This expands the interesting part of the curve in Figure 6 and shows more directly the relation between the maximum variability radius and performance. Note, in particular, that there is large potential payoff in performance for reducing the maximum variability radius a few percent more.

5 CONCLUSIONS AND FUTURE DIRECTIONS

The variability and performance results here are based on one subset of data for a particular plant, and are therefore not necessarily generalizable. Nevertheless, the design example shows that there is a potentially large performance cost in allowing for the effect of coupling variability on control-system stability, and that the selection of the coupling neutralization becomes very important to performance. With an arbitrarily chosen **N**, the performance is negligible, while with an optimal choice of **N** the performance increases to a respectable 8 dB. The goal in the analysis here is to gain insight into performance-stability tradeoffs as we develop a methodology for obtaining optimal system performance with real plant data.

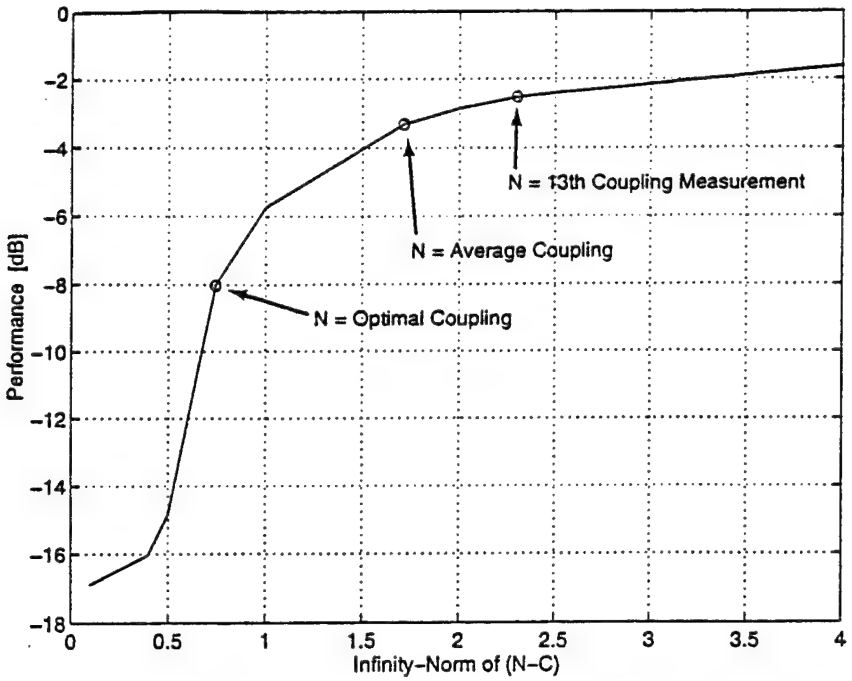

Figure 7. Control-system performance as a function of the maximum coupling variability radius. The data is the same as that in Figure 6.

By using MINCODE to optimize the choice of **N**, the nominal coupling, we have increased nominal performance as much as this straightforward approach will permit, while still guaranteeing control-system stability. To improve performance further we need a stability constraint that gives more freedom in the selection of **W**. An approach for accomplishing this was suggested in [1]. We can shrink the effective maximum variability radius by factoring the variability as

$$\delta \mathbf{C}_j = \mathbf{H}\,\Delta_j \mathbf{G} \tag{11}$$

where the pre- and post-multiplying matrices **H** and **G** are common to all the coupling data sets, and Δ_j is the "core" variability. The stability constraint analogous to Equation (6) then becomes

$$\|\mathbf{G}\,\mathbf{W}\,\mathbf{H}\|_\infty < \left(\max_j \|\Delta_j\|_\infty\right)^{-1} \qquad j = 1, 2, \ldots, \#\ \text{data sets} \tag{12}$$

Through judicious choosing of **G** and **H** we can make the core variability small and the right-hand side of Equation (12) large. This constraint ensures control-system stability while maximizing the freedom given to the weight matrix, **W**, in the objective-function minimization of Equation (5).

The matrices **G** and **H** will depend on the specific characteristics of the data set. In future work we will address how **G** and **H** might be selected for the CT-B (and other) data and how their selection will impact system performance. We will also consider the problem of how to determine directly **G**, **H**, and **N** via convex optimization.

This work has focused on the important tradeoff between nominal performance and stability, which is the so-called stability robustness problem. A related problem, not addressed in this report, is that of performance robustness. In this case we seek to design control laws that provide optimum performance, rather than maintain stability only, as plant model errors range over a given set. Work on this issue is ongoing.

SRI International

Technical Note • April 1996

COUPLING VARIABILITY FACTORIZATION FOR CONTROL-SYSTEM STABILITY ROBUSTNESS

Prepared by:

RICHARD P. HEYDT, *Research Engineer*
WILLIAM C. NOWLIN, *Senior Research Engineer*
Applied Control and Signal Processing Group
System Technology Division

Prepared for:

Office of Naval Research
Code 1222
800 N. Quincy St.
Arlington, VA 22217

Attn: Dr. Geoffrey L. Main
Program Manager, Structures

Contract N00014-94-C-0128

SRI Project 5796

A technical note is a working paper that presents the results of research related to a single phase or facet of a research problem. It presents the concepts, findings, and/or conclusions of the authors, alone, in fragmentary form intended for internal discussion and comment. Such notes are subject to revision, incorporation into more formal reports, or withdrawal.

Approved by:

J. RAUL MARTINEZ, *Director*
Applied Control and Signal Processing Group

Coupling Variability Factorization for Control-System Stability Robustness

Summary

In a previous technical note[1] we described how the coupling neutralization (**N**) in a neutralized feedforward control system could be selected to simultaneously guarantee stability and give good performance in the presence of variability. We showed with an example from the CT-B data that, by using MINCODE to determine the optimal **N**, we could improve a poorly performing system to one that has good performance (8 dB), while still ensuring that the system would be stable over the full range of variability represented by a set of coupling measurements. We also outlined an approach that would enable further performance gains by factoring the coupling variability. The methodology for this variability factorization has been developed but not yet implemented.

In this report we will illustrate a procedure for a simplified variability factorization and apply it to the same CT-B control-system problem. The simplified procedure gives excellent results: we obtain performance that is within a few dB of what can be achieved for non-robust design, while guaranteeing stability for all 24 measured coupling plants. The results demonstrate the power of the variability-factorization approach.

Controller-Design Example Problem

The example problem in the previous technical note was a 2-input, 2-output control system developed from the CT-B data set. Figure 1 is the standard Q-parameterized control-system block diagram. The problem is to design a controller that minimizes the responses at the performance sensors when the system is excited by an external disturbance. The design problem is "realistic" because we have to account for variability in the actuator-to-sensor coupling. Variability is inherent because the coupling is based on 24 measurements made at 1-hour intervals. The challenge, therefore, is to achieve good performance while making the system robust to changes in the coupling.

The control problem as formulated for solution with MINCODE is

$$\min_{\mathbf{W}} \left\| (\mathbf{E} + \mathbf{DWS})\, f_{PB} \right\|_\infty \tag{1}$$

subject to the *stability constraint*

$$\|\mathbf{W}\|_\infty < \left(\max_j \left\| \mathbf{N} - \mathbf{C}_j \right\|_\infty \right)^{-1} \qquad j = 1,\ 2,\ \ldots,\ 24 \tag{2}$$

and the *out-of-band enhancement constraint*

$$\left\| \mathbf{W} f_{OB} \right\|_\infty < \alpha \qquad (\alpha = 3) \tag{3}$$

The performance band filter f_{PB}. concentrates the minimization in the frequency range 1.0 to 3.5 KHz. Enhancement outside of the performance band is limited by the second constraint, which uses the out-of-band filter f_{OB}.

---

[1] Heydt, R.P., "Selection of Neutralization for Control-System Stability Robustness," SRI Applied Control and Signal Processing Group technical note, December 1995.

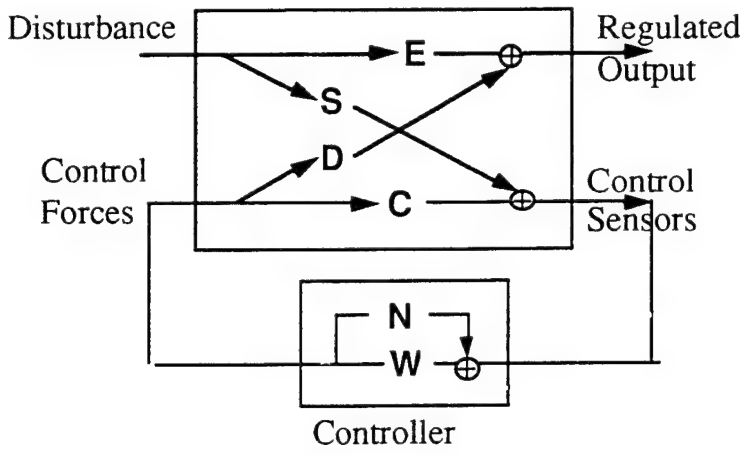

Figure 1. Q-Parameterized Control System

Most important is the first constraint (Equation (2)), which ensures system stability in the presence of additive variability in the coupling. We defined the quantity

$$\max_j \left\| \mathbf{N} - \mathbf{C}_j \right\|_\infty \tag{4}$$

as the *variability radius*. The radius is the maximum value, over all possible values of the coupling transfer function, of the infinity-norm of the difference between the selected neutralization, **N**, and the coupling, **C**. It is thus a measure of the variability observed over the 24 measurements of **C** in the CT-B data set. The stability constraint says that the infinity-norm of the weight set **W** must be less than one over the variability radius.

We showed in [1] that system performance depends strongly on the variability radius. Furthermore, we showed that by using the MINCODE engine to determine the **N** that minimizes the variability radius, we would get the best system performance for the problem *as stated* in Equations (1) through (3).

Variability Factorization

We can, however, achieve better performance while preserving system robustness to coupling variability. To do this we first develop a constraint that is less restrictive on **W** than Equation (2). Recall that Equation (2) is derived from the actual necessary and sufficient condition for control-system stability

$$\max_j \left\| \mathbf{W}\,(\mathbf{N} - \mathbf{C}_j) \right\|_\infty < 1 \tag{5}$$

If we factor the variability as

$$\mathbf{N} - \mathbf{C}_j = \mathbf{H}\,\Delta_j\,\mathbf{G} \tag{6}$$

where the pre- and post-multiplying matrices **H** and **G** are common to all measurement sets (independent of j), then the stability requirement can be written

$$\max_j \left\| \mathbf{G W H} \Delta_j \right\|_\infty < 1 \qquad (7)$$

Using the Cauchy-Schwarz inequality, we then formulate a new stability constraint that replaces Equation (2) in the controller design problem

$$\left\| \mathbf{G W H} \right\|_\infty < \left(\max_j \left\| \Delta_j \right\|_\infty \right)^{-1} \qquad j = 1, 2, .., 24 \qquad (8)$$

We have proposed a methodology[2] for determining (with the MINCODE engine) the factoring matrices **G** and **H** which give the best possible control-system performance for the variability in the measured coupling data. In the following section we show a simplified approach, based on frequency weighting, for selecting either **G** or **H**. As will be seen, this approach, while sub-optimal, can potentially yield excellent performance.

Frequency-Weighting Approach to **G**, **H** Selection

The reason that the stability constraint of Equation (2) leads to sub-optimal performance is that it over-constrains the weight set **W**. Equation (2) requires the infinity-norm of **W** to be limited by the worst-case variability, i.e., by the maximum deviation *over all frequencies* of the coupling transfer function **C** from the nominal coupling **N**. On the other hand, if we instead use Equation (8) and choose as **G** or **H** a suitable frequency-weighting function, we can avoid over-constraining **W**.

Figure 2 is a plot of the maximum singular values of the variability **N** – **C**, as a

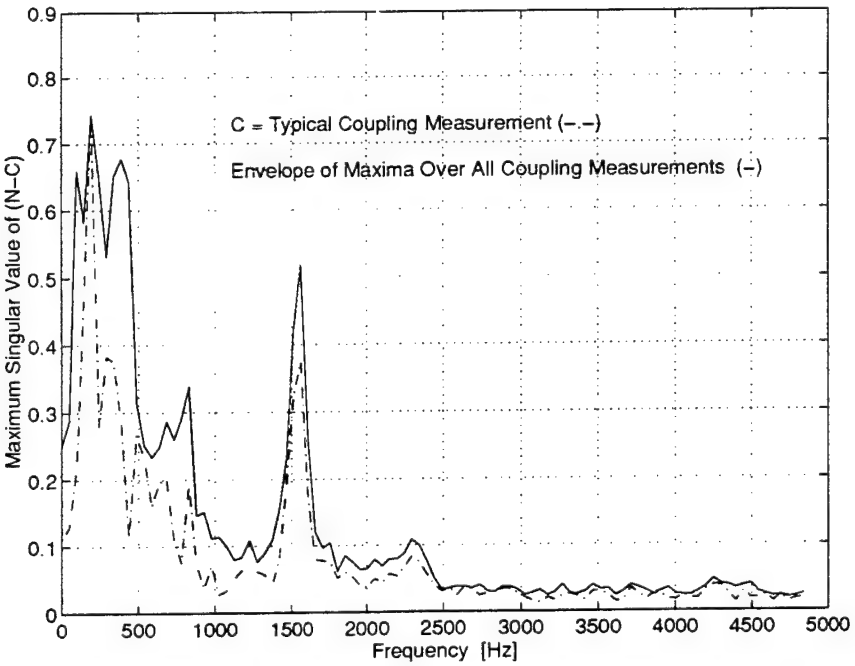

Figure 2. Maximum singular values of the variability **N** – **C** for two choices of the coupling **C**. **N** is the optimal neutralization transfer function.

---

[2] Guthart, G.S. and Titterton, P.J., "Variability Modeling for Robust Control Design," SRI Applied Control and Signal Processing Group client briefing, January 1996.

function of frequency, for the case in which **N** is the optimal neutralization (determined by MINCODE). The dashed line on the graph is the variability for a typical coupling measurement (the tenth of 24) from the CT-B data. The solid line is the envelope of the maximum variability over all 24 measurements of **C**. In other words, the magnitude of the observed variability at each frequency always lies at or below this line. Note that much of the variability occurs at frequencies below 1000 Hz, while the performance band for the control design problem is 1000 to 3500 Hz. We will take advantage of this in our design by applying a stability constraint of the form of Equation (8).

We choose the matrix **G** to be the identity matrix, and the matrix **H** to be

$$\mathbf{H} = \mathbf{I}\, h \tag{9}$$

where **I** is the 2 x 2 identity matrix and h is a filter with a spectrum equal in magnitude to the envelope of the variability measurements shown in Figure 2. The motivation for this choice of frequency-weighting follows from Equation (8). If **H** is the variability envelope (in the frequency domain), then the weight set **W** will be restricted in magnitude at those frequencies for which the maximum variability is large, but will be relatively unrestricted at frequencies for which the maximum variability is small. Depending on the particular control system and the measured data, this weighting will at worst have no effect on system performance, and at best have a very positive effect.

Control-System Performance

Figure 3 shows the MINCODE-generated control-system performance plot from [1]

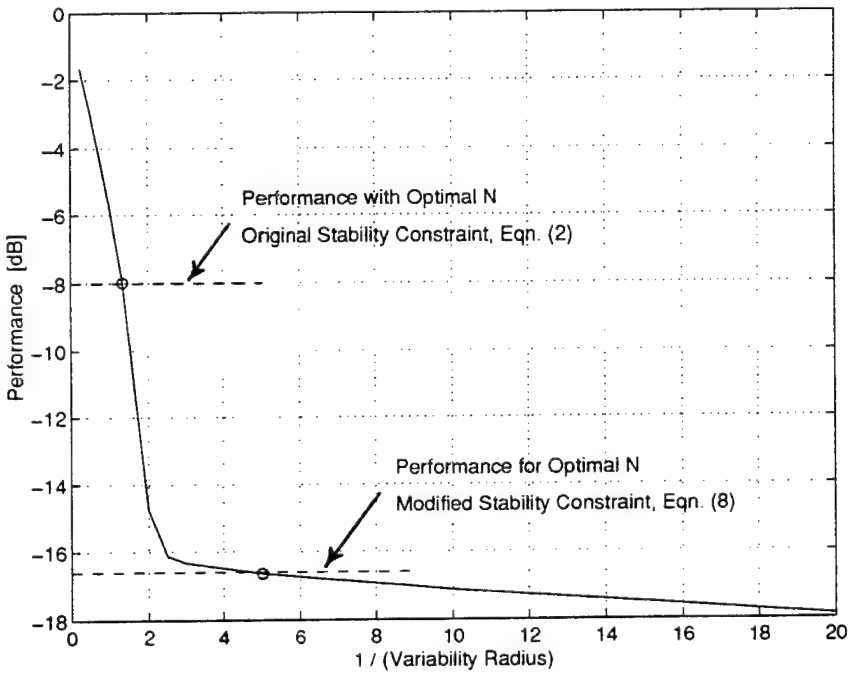

Figure 3. Control-system performance as a function of the inverse of the variability radius (solid line) with stability enforced by optimal frequency-independent means (Equation 2). 8 dB performance is obtained when the variability radius is minimized by selecting the optimal **N**. 16.5 dB performance is obtained if stability is enforced by a frequency-weighted criterion (Equation 8).

for the example problem as stated in Equations (1) through (3). Performance improves (becomes more negative) as the variability radius gets smaller (as we move to the right on the x-axis). We showed in [1] that by using the MINCODE engine to find the coupling that minimizes the variability radius, i.e., the optimal **N**, we could achieve 8.0 dB of performance. We can't reduce the radius further for the given coupling measurements, so 8 dB is the best possible performance as long as we enforce stability with Equation (2).

However, if we instead enforce stability with Equation (8), and use the frequency-weighting function **H** described above, we achieve a marked performance improvement to 16.5 dB. Looking at the curve in Figure 3, we see that this level of performance is approaching the best possible performance, the performance that would be obtained if there were *no coupling variability* (zero variability radius). By using the improved stability constraint, therefore, we achieve performance that with the original constraint would have been possible only if the variability radius were very small.

Conclusions

The approach described here efficiently guarantees control-system stability robustness, while minimizing degradation in performance due to coupling variability. This makes it extremely useful for designing real control systems, in which the transfer functions always have variability. We were able to obtain excellent performance (16.5 dB) in the CT-B example problem by intelligently selecting the frequency-weighting function used in the stability constraint. In future work, we will implement a more general, automatic approach, based on MINCODE optimization, for determining the matrices (**G**, **H**) that optimally factor any measured coupling variability.

Memo AC&SPG

TO: 5796 File

FROM: Gary Guthart

DATE: March 20, 1996

SUBJECT: Variability modeling

1 Summary

This memo describes progress in developing a variability model to be used in the MINCODE design methodology. By "variability model" we mean a way of representing a set of varying transfer functions, whose variations are known via measurement. These representations may be as complex as a complete ensemble of measurements representing all of the possible states of the system or as simple as a single scalar that bounds a norm over the entire range of possible variations. These representations are then used to describe the possible space of transfer function variations when computing a robust controller. Optimization procedures for the design of controllers given robustness requirements must rely on some model of the variability. As the detailed knowledge of the variability imbedded in the model increases, the performance of the controller also increases, since the optimization procedure in the design of the controller can be focused more tightly to be robust to just those parts of the plant that vary.

In what follows, we postulate a variability model that may be obtained using measured data. The solution for this variability model is the solution of a constrained, convex optimization, similar in form to the codes in the MINCODE engine. Next, we demonstrate how this variability model may be used in a controller design to guarantee robust stability to the observed variations. This approach is compared to a more conservative approach to stability robustness, resulting in a significant gain in performance while satisfying the same robustness criteria. The theory presented herein is developed for MIMO variability models. For simplicity, we exercised the algorithms on SISO problems as a test case. The extension of the numerical codes to the MIMO case is proceeding.

2 SISO Variability model

The variability model described herein is sufficiently general to handle the MIMO model matching problem, but the essential elements of the model's motivation are contained in the SISO case. Therefore, to get a feel for our definition of a variability model and its use, consider the following example. Consider a SISO disturbance rejection problem, shown in block diagram form in Figure 1. The control objective is to design the controller, K, such that the output signal, $y(t)$, is made small given any disturbance signal, $d(t)$.

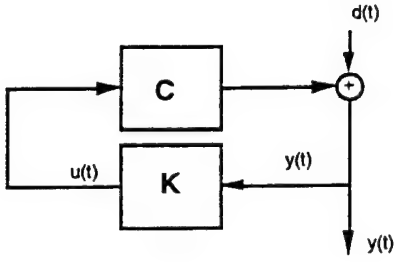

Figure 1: SISO sensitivity block diagram.

When the plant, C, is known and is linear, time-invariant (LTI), a number of optimal control algorithms can be used to design an LTI K. However, in real systems, C is rarely LTI, and identifications of C are contaminated by noise, nonlinearity and drift. The robust performance question, then, is to optimize and evaluate the performance of an LTI controller K in the presence of uncertain C.

We take the uncertain plant to be completely described by an ensemble of FIR transfer functions, C_j^n, where $j \in [1, 2, ..., N_e]$ and n indexes the taps of the FIR. Further, we assume that measurement noise (e.g., variations in the identification of C that are independent of input to the plant, $u(t)$) has been treated in the computation of the C_j^n via other methods (e.g., averaging). Note that these are significant assumptions. Determining how well an ensemble of plant measurements describes the total space of anticipated variations is a significant engineering task, involving estimates of the operating space of the plant and mechanisms of variability. Nonetheless, given the ensemble, C_j^n, that describes the possible variations of the plant we wish to control, we can proceed in modeling the variability and using the model in controller design.

To motivate the definition of our variability model, consider the stability robustness problem for our SISO plant. We proceed, as usual, in designing the controller K by representing the controller in its Q-parametrized form. Figure 2 shows the block diagram with the Q-parametrized controller. The requirement that the closed loop system be stable for all plants in the ensemble is given by the Small Gain Theorem:

$$\|Q(C_j - N)\|_\infty < 1 \quad j \in [1, ..., N_e] \tag{1}$$

where C_j are measured transfer functions, Q is the feedforward part of the controller that we wish to design and N is the controller's neutralization path, also available for us to design. A reasonable objective for designing the controller is to solve

$$\begin{aligned} \underset{Q}{\text{Argmin}} & \|F_{pb}(I - \bar{C}Q)\|_2 \\ & \|Q(C_j - N)\|_\infty < 1 \quad \forall\, j \in [1, ..., N_e], \end{aligned} \tag{2}$$

where $\bar{C}$ is a nominal representation of the plant and F_{pb} is a filter that directs the minimization to a particular performance band. In general, we would like to solve for

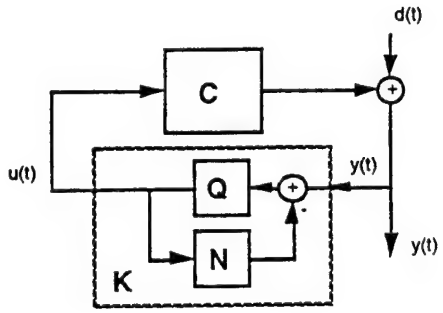

Figure 2: Q-parametrized controller for the SISO sensitivity problem.

Q and N simlutaneously.[1] Unfortunately, solution for Q and N simultaneously in this formulation is nonconvex in parameters for Q and N. Therefore, a suboptimal, two-step approach is adopted.

The two step approach for solving for the neutralization and the forward weights is as follows. From Equation 2, we see that Q is constrained by the decoupling error to satisfy the robust stability requirement. Using the Cauchy-Schwartz Inequality,

$$\|Q(C_j - N)\|_\infty \leq \|Q\|_\infty \|(C_j - N)\|_\infty. \tag{3}$$

Therefore, constraining

$$\|Q\|_\infty \|(C_j - N)\|_\infty < 1 \quad \forall\, j$$

will satisfy the robust stability requirement. Consequently, the first step in finding the controller is to find N such that

$$\underset{N}{\text{Argmin}} \sup_j \|(C_j - N)\|_\infty = D. \tag{4}$$

This minimization for N is convex in the parameters for N and thus, may be solved unambiguously for N using linear programming techniques developed for MINCODE. The second step of the controller design process, then, is to find the forward weights, Q, satisfying

$$\underset{Q}{\text{Min}} \|F_{pb}(I + \bar{C}Q)\|_2 \qquad \text{subject to}$$

$$\|Q\|_\infty D < 1 \quad \text{which implies} \quad \|Q\|_\infty < \frac{1}{D}.$$

[1]The selection of the nominal plant over which the optimization is to take place is a part of the broad performance robustness problem. Typical, though clearly suboptimal ways of choosing $\bar{C}$ include taking $\bar{C}$ to equal a particular plant C_p, an average plant C_{avg} or a plant computed from the ensemble in some other way (e.g., N). Furthermore, the closed loop transfer function in the presence of decoupling error is the nonlinear functional

$$H = I + C(I + \Delta CQ)^{-1}Q$$

where $\Delta C = C_j - N$. This functional is clearly nonlinear in Q, making the optimization for Q nonconvex and motivating the simplification to the closed loop cost function affine in Q, $I + \bar{C}Q$.

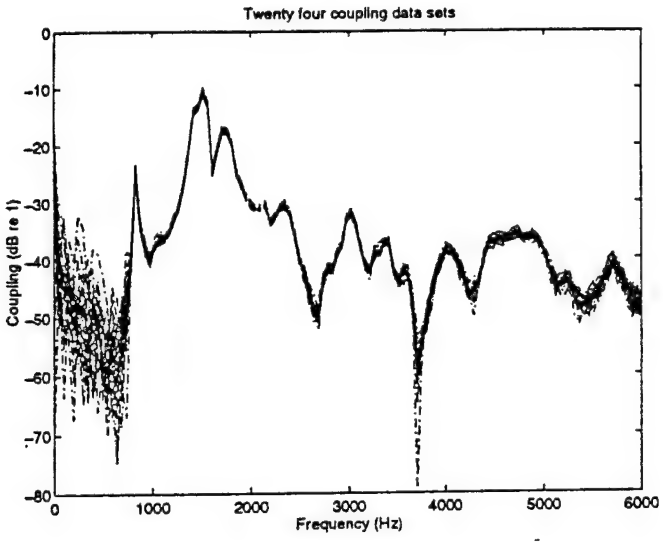

Figure 3: Spectra of plant ensemble for numerical example described in the text.

This equation for Q is convex in the parameters for Q and may also be solved using MINCODE.

A problem with this approach is that the constraint on Q may be overly conservative when the data contains most of its variability outside of the performance band. The following numerical example will help to illustrate this point. Using data from the CT-B model, we take the coupling from one on-board actuator to one on-board ring sensor to be our plant. We have twenty-four measurements of the same plant, each taken one hour apart. The spectra of this ensemble is shown in Figure 3, with the frequency axis focused on the low frequency portion of the spectrum. In this example, the variability in the plant is quite large below 1 kHz, as evidenced by the thickness of the envelope of transfer functions. Figure 4 shows the optimal neutralization, N found by solving System 4 along with the envelope of the measured plants. Note that N need not stay within this envelope because the envelope shows only magnitude information. If the phase of each measured plant set differs widely, the optimal neutralization will move outside of the envelope since the infinity norm is on a difference of complex functions.

If we choose a performance band for our disturbance rejection problem to be from 1500 to 2000 Hz, then the variability in the plant will be out-of-band. Using the conservative approach outlined above, we constrain all frequencies of Q to have infinity norm less than D, the worst case norm anywhere in the system from 0 frequency to Nyquist. This results in a constraint that is too strict in the performance band and results in reduced system performance, as we will show below.

A clear approach to solving this conservativeness problem is to allow for a frequency dependent bound on the weights, Q. The question, then, is how to choose the bound. We can trace the conservativeness of the above stability robustness constraint to the use of the Cauchy-Schwartz inequality. The physical condition that needs to be satisfied to

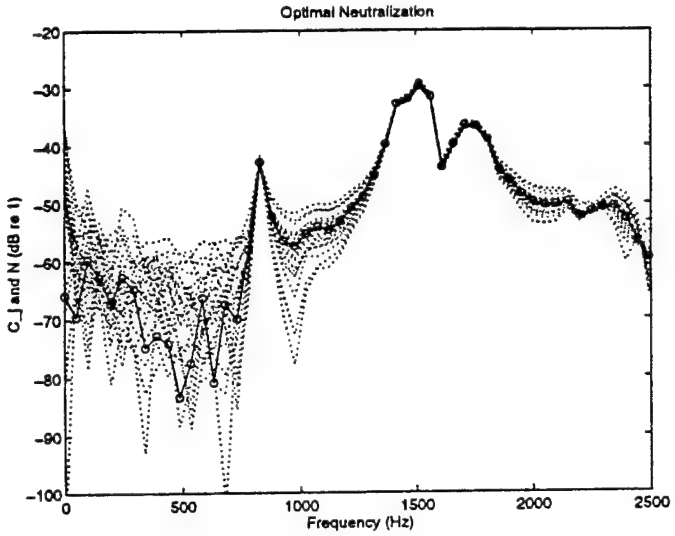

Figure 4: Spectrum of the optimal neutralization.

guarantee stability over our ensemble is

$$\|Q(C_j - N)\|_\infty < 1$$

but the actual condition we satisfy is

$$\|Q\|_\infty D < 1.$$

If we can bound $(C_j - N)$ as a function of frequency,

$$|C_j(\omega) - N(\omega)| \leq G(\omega) \quad \text{for } \omega \in [0, \text{Nyquist}] \quad \text{and } \forall\, j \tag{5}$$

then, since in the SISO case the infinity norm is just the maximum of the operator in the frequency domain and since the operator composition (convolution) is just frequency by frequency multiplication, we can write

$$\|Q(C_j - N)\|_\infty \leq \|QG\|_\infty \leq \|Q\|_\infty D, \tag{6}$$

which shows that $\|QG\|_\infty$ bounds our physical stability condition and is less conservative than our first attempt. Therefore, bounding

$$\|QG\|_\infty < 1 \tag{7}$$

clearly satisfies the small gain theorem. Indeed, in the SISO case, given N found by the optimization in System 4, $G(\omega)$ is easy to obtain. Since C_j and N are known, $G(\omega)$ is simply

$$G(\omega) = \max_j |C_j(\omega) - N(\omega)| \tag{8}$$

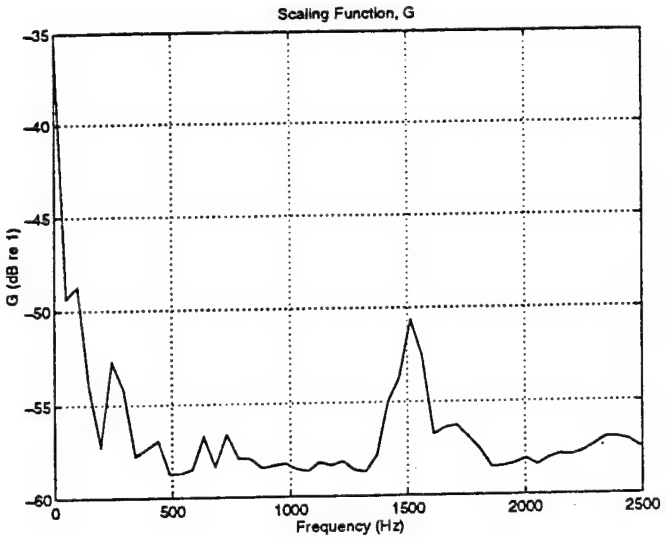

Figure 5: $G(\omega)$ for the neutralization and coupling shown above.

taken frequency by frequency. For the numerical example given above, G is shown in Figure 5.

We can illustrate the effect of this reduction in conservativeness of the stability robustness bound on performance in our numerical example. Our conservative approach to the controller design problem solves

$$\begin{aligned} \underset{Q}{\text{Min}} \; & \|F_{pb}(I + C_1 Q)\|_2 \qquad \text{subject to} \\ & \|Q\|_\infty \; < \; \frac{1}{D}, \end{aligned} \tag{9}$$

while the less conservative approach solves

$$\begin{aligned} \underset{Q}{\text{Min}} \; & \|F_{pb}(I + C_1 Q)\|_2 \qquad \text{subject to} \\ & \|QG\|_\infty \; < \; 1. \end{aligned} \tag{10}$$

Taking F_{pb} to be a band-pass filter selecting the frequency range 1500 to 2000 Hz in our data set, the approach in Equation 9 gives a nominal performance of X dB while the less conservative approach in Equation 10 improves performance to Y dB. Both designs are guaranteed to be stable to all of the coupling variations in the ensemble. The second design uses the more detailed information contained in G to improve nominal performance.

As a last remark on our SISO example, we address the performance sensitivity question. Above we examined nominal performance as a function of the stability constraint. What happens to actual performance as the plant varies across the ensemble? Although we have not designed the above controllers to minimize performance sensitivity, we hope

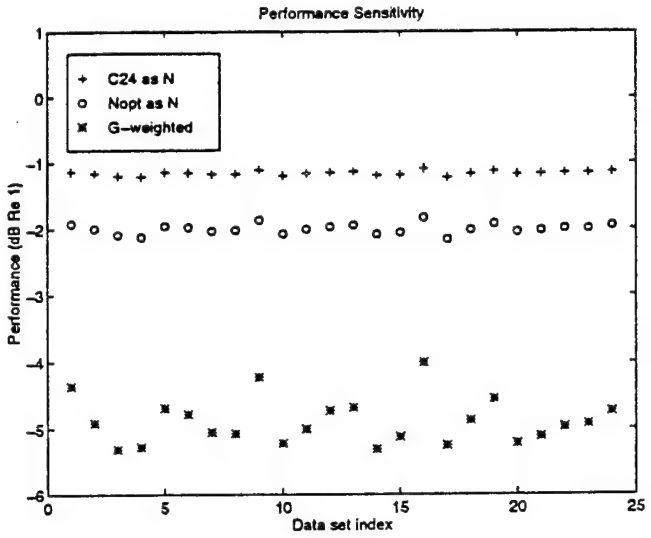

Figure 6: Performance for the direct constraint with C_{24} chosen as N (+) with the optimal N computed (o) and the G weighted constraint (*) while the plant moves through the twenty-four data sets.

that our designs exhibit low sensitivity in performance. Figure 6 shows the actual performance of the controllers as the plant moves to each of the twenty-four data sets. This figure shows that the performance gains realized by using G remain beyond just nominal performance, giving reasonably low performance sensitivity. Of course, this result is in no way general, but it points the way towards including the variability model (in this case G) into minimization criteria that include performance sensitivity.

3 MIMO Variability Model

We now return to the more general MIMO model matching problem. Figure 7 shows the now familiar block diagram of a Q-parametrized controller in the model matching case. Now we take C_j to be the measured, physical coupling between the actuators and sensors, a transfer matrix having N_a columns and N_s rows where N_a is the number of actuators and N_s the number of sensors. As usual in the MINCODE paradigm, the time dynamics of the transfer functions are approximated by FIR taps. The controller design problem assuring stability robustness over an ensemble of C_j, $j \in [1, 2, ..., N_e]$ is

$$\begin{array}{rcl} \underset{Q}{\text{Min}} & & \|E + DQS\| \\ \|Q(C_j - N)\|_\infty & < & 1 \quad \forall\, j, \end{array} \tag{11}$$

where the constraint is the Small Gain Theorem, once again. The first approach for simplifying the constraint, described in the SISO example above, is easily generalized to

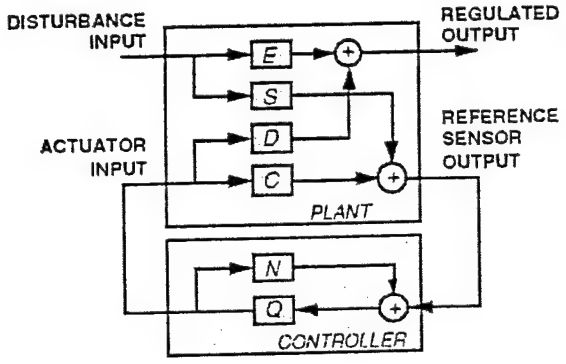

Figure 7: Block diagram for the Q-parameterized model matching problem.

the MIMO case,

$$\|Q(C_j - N)\|_\infty \leq \|Q\|_\infty \|(C_j - N)\|_\infty \leq \|Q\|_\infty D < 1,$$

where the infinity norm now operates on matrix transfer functions and N is chosen to minimize D,

$$\underset{N}{\text{Argmin}} \sup_j \|C_j - N\|_\infty = D. \tag{12}$$

Generalizing the frequency weighting, $G(\omega)$, that generates a convenient and less conservative constraint on the weights is more difficult in the MIMO case. The following optimization based technique has been developed.

Again using the stability robustness constraint as motivation for a variability model, we define

$$\Delta C_j = C_j - N, \tag{13}$$

which is the decoupling error operator for each data set in our ensemble. The modeling challenge is to represent this new ensemble (ΔC_j) in a way that is convenient to use in our MINCODE design methodology. In the SISO case, we defined a $G(\omega)$ that was the frequency domain envelope of the decoupling error. In the MIMO case, we generalize this concept to include matrix operators. We start by introducing the following decomposition of the decoupling error,

$$\Delta C_j = G \Delta_j H, \tag{14}$$

where G and H are independent of j but are frequency dependent. Note that there is no loss of generality in this decomposition. The motivation for this decomposition is as follows: In the SISO case we "captured" the shape of the variability in the decoupling

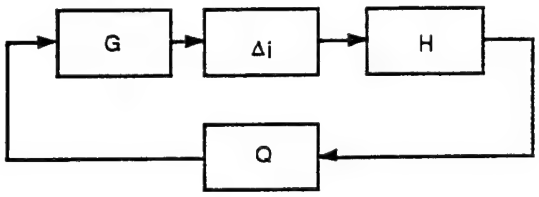

Figure 8: Controller feedback loop.

error by bounding the ensemble. Here, H and G capture the input dynamics and the output dynamics in the decoupling error. Of course, there is no guarantee, *a priori*, that this decomposition will be effective. Its power remains to be determined by application, shown below.

The decomposition of ΔC_j in Definition 14 amounts to modeling each element of the coupling ensemble,

$$C_j = N + G\Delta_j H, \tag{15}$$

so that the closed loop in the controller reduces to the loop shown in Figure 8. The condition that this loop be stable may be written as

$$\|HQG\Delta_j\|_\infty < 1 \quad \text{for all } j. \tag{16}$$

Note that there has been no approximation to this point. Now, we seek a condition on Q that satisfies the above equation and is independent of j, the ensemble index. As usual, we can write

$$\|HQG\Delta_j\|_\infty \leq \|HQG\|_\infty \|\Delta_j\|_\infty.$$

Now, so long as Δ_j is stable (which it is for FIR models), we are free to choose a bound for $\sup_j \|\Delta_j\|_\infty$. That this is so can be seen from Equation 14, where any scalar multiplier may be absorbed between G, H and Δ_j. For convenience, bound $\|\Delta_j\|_\infty < 1$. Therefore, we have

$$\|HQG\Delta_j\|_\infty \leq \|HQG\|_\infty, \tag{17}$$

and requiring that Q satisfy

$$\|HQG\|_\infty < 1 \tag{18}$$

will guarantee that the loop is stable. Thus far we have not specified how to choose H and G from the measured data C_j; we have only required that $\|\Delta_j\|_\infty < 1$. Intuitively, we know that allowing H and G to be large will constrain Q to be small, via Equation 18. Therefore, we seek to minimize the "size" of both H and G while enforcing $\|\Delta_j\|_\infty < 1$.

Finally, we propose the following procedure for finding H, G and N. If we define the size of H and G to be given by

$$\|R_{gg} + R_{hh}\|_2, \quad \text{with } R_{gg} = G^*G \; R_{hh} = H^*H,$$

then we can find H, G and N by solving the following constrained optimization

$$\min_{R_{gg}, R_{hh}, N} \quad \|R_{gg} + R_{hh}\|_2$$

$$\begin{bmatrix} R_{gg} & (C_j - N) \\ (C_j - N)^* & R_{hh} \end{bmatrix} > 0 \quad \forall\, j. \tag{19}$$

The positive definite constraints above are both necessary and sufficient to guarantee that $\|\Delta_j\|_\infty = \|G(C_j - N)H\|_\infty < 1$, which can be shown using the Schur complement of the compound matrix in Equation 19. The above constrained minimization is convex in FIR parameters of N, R_{gg} and R_{hh}. One can recover the transfer functions G and H from R_{gg} and R_{hh}. With G and H in hand, the optimization for the Q-parameter that guarantees stability robustness to the varying plant, C_j is

$$\min_{Q} \quad \|E + DQS\| \quad \text{subject to}$$

$$\|GQH\|_\infty < 1.$$

When reduced to the SISO case discussed at the outset of this memo, the above procedure for finding G yielded the same function as the heuristic described above.

4 Conclusions and future work

We believe that the approach to modeling variability described above is entirely novel. The results for the SISO case, both in finding the model function G and its use in finding the Q parameter are extremely encouraging and point the way towards controller designs capable of operating well in real-world environments. The novelty of the approach leaves several issues open to future research, necessary to fully characterize the computation and optimality of the modeling procedure. These issues include

- Serial versus parallel computation of H, G and N. Since the programs required to solve the constrained optimizations are computationally intensive, it is worth determining whether N may be determined by an independent optimization relative to G and H. Early results indicate that computation of N minimizing $\|C_j - N\|_\infty \; \forall\, j$ and finding G and H subsequently yields solutions that are quantitatively equivalent to the parallel approach. This early work does not constitute a proof, and further study is needed.

- Time-domain versus frequency-domain computation. Because of the structure of the FIR representation of the coupling operator, computations of the operators G, H and N are very simple when computed frequency-by-frequency. However, frequency-by-frequency computations do not naturally enforce causality. We know that N must be causal, since we implement it within the controller. I suspect that G and H need not be causal, since they are always used as frequency domain operators, but this point has not yet been proved, as it should be. Early numerical experiments have shown that neither time-domain nor frequency-domain identification of N give perfectly causal N. Our suspicion is that sampling errors in

approximation of the infinity norm are the cause of acausal leakage, but again, this issue needs to be fleshed out.

- The definition of the size of G and H is somewhat arbitrary and one could postulate different norms that measure the size of G and H. The effect of these different norms on the solutions for G and H should be investigated.
- The models described here have only been used to evaluate performance while constraining loop gain. Can these models be used to help improve performance sensitivity?

Many thanks to Richard Heydt, Bill Nowlin, Julia Olkin and Paul Titterton for the opportunity to work on this project.

March 28, 1995

Christopher E. Ruckman
Naval Surface Warfare Center
Carderock Division
Code 725

Dear Chris,

Thank you for inviting SRI to attend your meeting with BBN on 20 March 1995 at NSWC Carderock. This memo summarizes SRI's recommendations for the Cradle experiments, as requested at the end of that meeting.

In reviewing the Program's opportunities for Cradle experiments and associated needs, we envision the following set of hypotheses to test during this experiment.

- Is driving a component of the transmitted force to a small value at each mount point sufficient to reduce far-field radiation?
- Can a single-input, single-output control law applied at each mount robustly reduce transmitted vibration?
- Are actuators located below-mount superior for control than those located above-mount?
- How does the truss oscillate when driven by representative truss-mounted machinery?
- How does that oscillation couple to the outer structure to result in far-field radiation?
- What shaker orientation, or combination of shakers, most effectively duplicates the transmission of energy to the outer structure?

To answer these questions we believe would require a significant amount of instrumentation beyond that which is currently planned. The correct way to approach these questions is through the formulation of a good model for analysis; such a model is currently unavailable and thus we cannot eliminate particular types and locations of actuators.

A crucial, and to our knowledge, unresolved issue that forms the underpinning to any analysis of the above is coupling:

> In the matrix of transfer functions relating control actuation to performance sensed ("coupling"), what is the relative level of off-diagonal terms compared to that of the diagonal terms?

SRI International
333 Ravenswood Ave. • Menlo Park, CA 94025 • (415) 326-6200 • TWX: 910-373-2046 • Telex: 334486 • Facsimile: (415) 326-5512

Unfortunately, the coupling issue is also pertinent to our study of the physical mechanisms responsible for propagation of vibration from the truss to the outer structure and the far field. Since we do not know the energy transmission mechanism, we do not know, a priori, what orientation to choose for actuation (circumferential, axial, radial).

At last Monday's meeting we had discussed placing multiple shakers on a single, particular mount. Our goal was to determine these physical mechanisms, and to determine an actuation approach that drives the same mechanism. However, if the off-diagonal coupling terms are non-negligible, these determinations cannot be made by actuating a single mount. The response at any performance sensor will be a combination of contributions from all mounts, without a full set of calibrated actuation responses.

Because of the uncertainties regarding coupling and transmission mechanisms, we believe that the truss should *ideally* be instrumented with a full set of actuators at each mount. Here, a "full set" would include three orthogonal force actuators above the passive mount and three more below, plus three orthogonal torque actuators both above and below the passive mount. This is obviously unrealistic.

Specific Recommendations

Shaker Locations and Orientations

The schedule and cost constraints force us to suggest a compromise recommendation that will not test all the hypotheses above, but will yield useful information. Specifically, we suggest one fully actuated mount and nine singly actuated mounts (we assume that a tenth actuator will be one of the actuators on the fully actuated mount) on one side of the axis of symmetry. We recommend circumferential actuators on the nine singly-actuated mounts, based on best guess of how the mount will flex under load to launch waves. Circumferential actuation is also likely the easiest for installation.

If the system is sufficiently decoupled to enable SISO measurements, then the mount with multiple actuators will yield data on appropriate actuator components for matching truss excitations. On the other hand, if the system cannot be decoupled, then we will obtain one complete set of MIMO transfer functions, which will establish the coupling level for one type of actuation, and will enable us to match truss excitations with actuation patterns.

To summarize, we suggest the placement of three actuators on the primary mount and one each on the remaining nine mounts on that same side. The three actuators

on one mount would actuate in circumferential, axial, and radial directions, and the other nine would be circumferential. We realize this does not span the full set of forcing degrees of freedom, as we are ignoring torque actuation. However, this is a reasonable compromise recommendation; it yields important information in the case when cross-coupling is negligible and also in the case when it is not.

Shaker Positioning Relative to Passive Mount

We suggest that the actuators be attached below-mount, as outlined by BBN, because physical constraints appear to drive this selection. Our suggestion under ideal circumstances would be to shake both above- and below-mount.

The issue of shaker location—either above the passive mount or below it—arose during Monday's meeting. Again, we believe this issue should be resolved with the aid of a suitable physical model; at this time only simple spring-mass-dashpot-type models have been available for our analysis. While it is clear that a below-mount actuator will generally result in lower command force requirements, an above-mount actuator may result in a control law that is less sensitive to plant modeling errors. Finally, we are intrigued by BBN's assertion of a simplification of the plant when below-mount force actuation with force feedback is used, but do not understand the modeling assumptions on which this is based.

Sensor Locations

We are assuming that each mount point will be instrumented above and below the passive mount with tri-axial accelerometers, and that responses at these sensors resulting from actuators described above will be available.

Instrumentation of the mounts with force sensors, would also be beneficial. Given the time constraints, we defer to NSWC to choose the appropriate sensor type (PVDF, Kistler, etc.).

Other Considerations

The actuator selector switch discussed in Monday's meeting is a good idea in light of our recommendation above to install 12 actuators on the mounts. We understand the difficulty in making a reliable selector switch in a short time.

Frequency range: The standard concerns for control law design apply here, i.e., good signal-to-noise for a decade above and below the control band. As was mentioned in the meeting, someone familiar with active control simulation should design the waveforms.

Conclusions

We appreciate the efforts of NSWC in instrumenting the model and making these measurements, as well at those of BBN for their useful and constructive comments. We hope that the issues of cost and accessibility can be resolved in light of the tight schedule, and that our suggestions are helpful to that end.

Best wishes,

Bill

William C. Nowlin
(415) 859–5628

Richard P. Heydt
(415) 859-4452

David S. Flamm

cc: Geoffrey L. Main, ONR
Arch Owen, BBN